AD 744505

AD

USAAMRDL TECHNICAL REPORT 72-17

A TETRA-CORE STRESS ANALYSIS MODEL

By

Alan Lee Dobyns
David C. Jack

April 1972

EUSTIS DIRECTORATE
U. S. ARMY AIR MOBILITY RESEARCH AND DEVELOPMENT LABORATORY
FORT EUSTIS, VIRGINIA

CONTRACT DAAJ02-71-C-0056
THE BOEING COMPANY
SEATTLE, WASHINGTON

Approved for public release;
distribution unlimited.

261

DISCLAIMERS

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the US Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission, to manufacture, use, or sell any patented invention that may in any way be related thereto.

Trade names cited in this report do not constitute an official endorsement or approval of the use of such commercial hardware or software.

DISPOSITION INSTRUCTIONS

Destroy this report when no longer needed. Do not return it to the originator.

DEPARTMENT OF THE ARMY
U. S. ARMY AIR MOBILITY RESEARCH & DEVELOPMENT LABORATORY
EUSTIS DIRECTORATE
FORT EUSTIS, VIRGINIA 23604

This study was performed under Contract DAAJ02-71-C-0056 with The Boeing Company, Seattle, Washington. The technical monitor for this contract was Mr. I. E. Figge, Structures Division.

This report contains the results of a study to develop a three-dimensional finite-element computer program for "Tetra-Core" elements.

This report has been reviewed by the Eustis Directorate, U.S. Army Air Mobility Research and Development Laboratory and is considered to be technically sound. It is published for the exchange of information and the stimulation of future research.

TASK 1F162203A17003
Contract DAAJ02-71-C-0056
USAAMRDL Technical Report 72-17
April 1972

A TETRA-CORE STRESS ANALYSIS MODEL

D180-14257-1

By

Alan Lee Dobyns
David C. Jack

Prepared by

The Boeing Company
Seattle, Washington

for

EUSTIS DIRECTORATE
U. S. ARMY AIR MOBILITY RESEARCH AND DEVELOPMENT LABORATORY
FORT EUSTIS, VIRGINIA

Approved for public release;
distribution unlimited.

ABSTRACT

The results of a five-month analytical program are presented, describing a stress analysis method for a three-dimensional fiber composite structure. The intersections of the elements or webs of this structure form polyhedra in the form of triangular prisms, Tetra-core, and truncated forms of these.

The finite-element model generates the elements and polyhedra automatically with an optimization routine which selects the minimum weight structure that will meet the stiffness and loads requirements.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) The Boeing Company Seattle, Washington	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP
3. REPORT TITLE A TETRA-CORE STRESS ANALYSIS MODEL	
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final Report	
5. AUTHOR(S) (First name, middle initial, last name) Alan L. Dobyns David C. Jack	

6. REPORT DATE April 1972	7a. TOTAL NO. OF PAGES 258	7b. NO. OF REFS 8
8a. CONTRACT OR GRANT NO. DAAJ02-71-C-0056 b. PROJECT NO. Task 1F162203A17003 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) USAAMRDL Tech Report 72-17 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) D180-14257-1	
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Eustis Directorate, U.S Army Air Mobility Research and Development Laboratory, Fort Eustis, Virginia	

13. ABSTRACT

The results of a five-month analytical program are presented, describing a stress analysis method for a three-dimensional fiber composite structure. The intersections of the elements or webs of this structure form polyhedra in the form of triangular prisms, Tetra-core, and truncated forms of these.

The finite element model generates the elements and polyhedra automatically with an optimization routine which selects the minimum weight structure that will meet the loads requirement.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Tetra-Core						
Fiber Composite						
Stress Analysis						
Structural Optimization						

FOREWORD

This report was prepared by The Boeing Company, Seattle, Washington, under U. S. Army Contract DAAJ02-71-C-0056 (DA Task 1F162203A17003), and contains an analytical stress analysis model of a three-dimensional fiber composite, Tetra-Core. The contract was started 28 May 1971 and was completed 17 December 1971. Project Engineer for this program was Mr. I. E. Figge, Sr., Structures Division, Eustis Directorate, U. S. Army Air Mobility Research and Development Laboratory.

The authors acknowledge the following Boeing personnel for their support: David L. Beste and Robert Blaisdell.

Preceding page blank

TABLE OF CONTENTS

Page

ABSTRACT.. iii

FOREWORD.. v

LIST OF ILLUSTRATIONS..................................... ix

INTRODUCTION.. 1

MODEL DESCRIPTION... 7

INPUTS.. 7

AUTOMATIC MODEL GENERATOR................................. 22

Flat Plate Nodal Generator................................ 22
Cylinder Nodal Generator.................................. 23
Airfoil Nodal Generator................................... 23
Plate Element Generator................................... 24
Flat Plate.. 24
Cylinder and Airfoil...................................... 25
Flat Plate Loads.. 25
Cylinder Loads Generation................................. 28
Airfoil Loads Generation.................................. 29
Nodal Fixity.. 29

FINITE ELEMENT STRESS ANALYSIS............................ 31

Linear Strain Element..................................... 31
Quadrilateral Element..................................... 39
Constant Strain Triangle.................................. 39
Consistent Surface Loads Vector........................... 40
Calculation of Deflections................................ 41

OPTIMIZATION.. 45

NONLINEAR ANALYSIS.. 53

BUCKLING ANALYSIS... 57

MARGIN OF SAFETY.. 58

CALCULATION OF ULTIMATE LOADS............................. 59

CALCULATION OF STIFFENESSES............................... 59

OUTPUT DESCRIPTION.. 61

SAMPLE PROBLEMS... 65

COMPARISON WITH UNIVERSITY OF KANSAS TESTS................ 69

MODEL CONSTRAINTS... 70

COMPUTER PROGRAM.. 71

LIST OF SUBROUTINES....................................... 90

Preceding page blank

TABLE OF CONTENTS (Cont)

	Page
DESCRIPTION OF VARIABLES USED IN COMMON	92
DESCRIPTION OF VARIABLES USED IN SUBROUTINES	95
LITERATURE CITED	178
GLOSSARY	179
APPENDIXES	181
I. ADDITIONAL GENERATION EXAMPLES	181
II. EXAMPLE OF COMPUTER PROGRAM OUTPUTS	204
DISTRIBUTION	251

LIST OF ILLUSTRATIONS

Figure		Page
1	Typical Applications of Tetra-Core	2
2	Basic Tetra-Core Element	3
3	Sequence of Operation in Tetra-Core Analysis Program	6
4	Basic Tetrahedron Geometry	8
5	Tetra-Core Input Geometry	9
6	Material Properties Required as Input	11
7	Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	12
8	Nodal Numbering System, Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	13
9	Skew A Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	14
10	Skew B Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	15
11	Skew A Plates in Cylinder (Computer Plot)	16
12	Flat Plate Applied Loads	18
13	True Tetra-Core Cylinder With Torsion Load Applied, $L_X = 10$	20
14	True Tetra-Core Airfoil With Torsion Load Applied, $L_X = 20$	21
15	Structural Stiffness Matrix	42
16	Design Space	46
17	Two-Dimensional Optimization Example	48
18	Composite Constraint Gradient	48
19	Constraint Surfaces of Three-Dimensional Problem	50
20	Global Optimum	50

LIST OF ILLUSTRATIONS (Cont)

Figure		Page
21	Example of Tetra-Core Optimization	52
22	Method Used To Determine Moduli for Plastic Analysis	54
23	Example of Nonlinear Analysis	56
24	Elements Used in Tetra-Core Program	61
25	Added Mid-Side Nodes, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$	62
26	Input Data for Nonlinear Analysis	66
27	Input Data for Airfoil Analysis	67
28	Input Data for Optimization Run	68
29	Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$	182
30	Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 20$, $L_Y = 6$	183
31	Nodal Numbering System, Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 6$	184
32	Nodal Numbering System, Truncated Tetra-Core Flat Plate, $L_X = 8$, $L_Y = 4$	185
33	Vertical Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4$, $L_Y = 4$	186
34	Skew A Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4$, $L_Y = 4$	187
35	Skew B Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4$, $L_Y =4$	188
36	Skew A Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$	189
37	Skew B Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$	190
38	Load N_X Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	191

LIST OF ILLUSTRATIONS (Cont)

Figure		Page
39	Load N_Y Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	192
40	In-Plane Shear Load N_{XY} Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	193
41	Moment M_X Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	194
42	Moment M_Y Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	195
43	Twisting Moment M_{XY} Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$	196
44	Vertical Shear Load XQSHR Applied to Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	197
45	Vertical Shear Load YQSHR Applied to Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	198
46	Load N_X Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	199
47	Load N_Y Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	200
48	In-Plane Shear Load N_{XY} Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	201
49	Moment M_X Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	202
50	Moment M_Y Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$	203
51	Output From Optimization Run	205

INTRODUCTION

This document contains a description of a three-dimensional finite-element stress analysis computer model developed to provide a tool for defining the geometry of Tetra-core structures. The model uses material properties and applied loads in determining the final composite configuration. The primary polyhedron shape is a tetrahedron; one form of the model can produce a Tetra-core, where all elements intersect to form an equilateral tetrahedron.

The objective of the program was to develop a finite-element stress model that would provide a method of analysis of a complex nonhomogenous structure. The model was developed to run on the IBM 360/44. The size of problem that can be run is limited only by the number and size of high-speed discs available.

Tetra-core is a new construction concept developed for use on helicopters, airplanes, and other vehicles where lightweight and damage-tolerant structures are desirable. A typical application is shown in Figure 1. It is made by filament winding fiber glass or advanced composite fibers into an interlocking group of tetrahedrons, as shown in Figure 2. This arrangement appears to give a lightweight, truss-type structure that will be able to sustain damage and still carry a load, because of the high degree of structural indeterminancy. Tetra-core is constructed using a standard filament-winding machine. The fibers are laid in legs oriented in one direction, then the fibers of the next direction are laid over them, thus providing good load transfer between legs. Adhesive is applied to the fiber as it is being wound. The final composite is then ready for the curing process.

Several stress analysis methods have been considered for use with Tetra-core. A simple stress analysis, in which the load in each leg is resolved into its components by a summation of forces, does not account for the structural indeterminancy of Tetra-core, and will probably give conservative results.

The classical laminated plate theory widely used with boron and graphite-layered panels does not apply to Tetra-core since it is not a homogenous material. Each leg of a Tetra-core can be analyzed using laminated plate theory, but a method that accounts for the redundancy caused by the interweaving of the legs should be used in determining the strength of a Tetra-core panel.

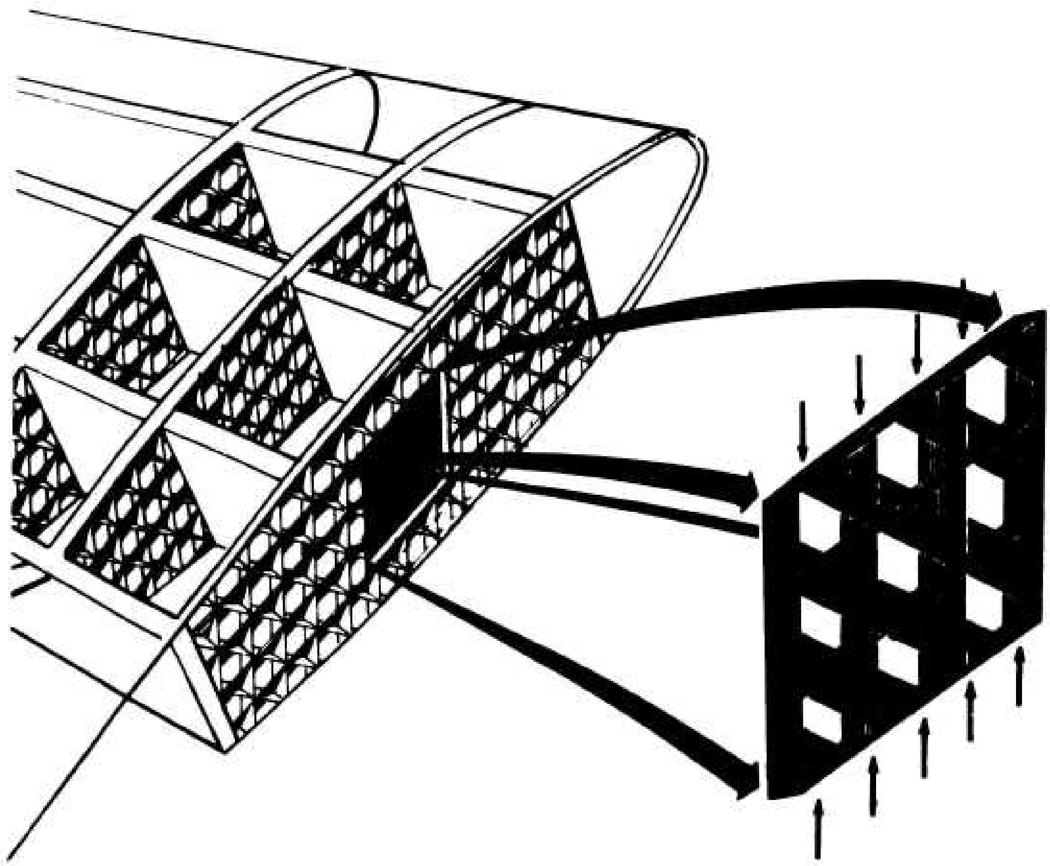


Figure 1. Typical Application of Tetra-Core.

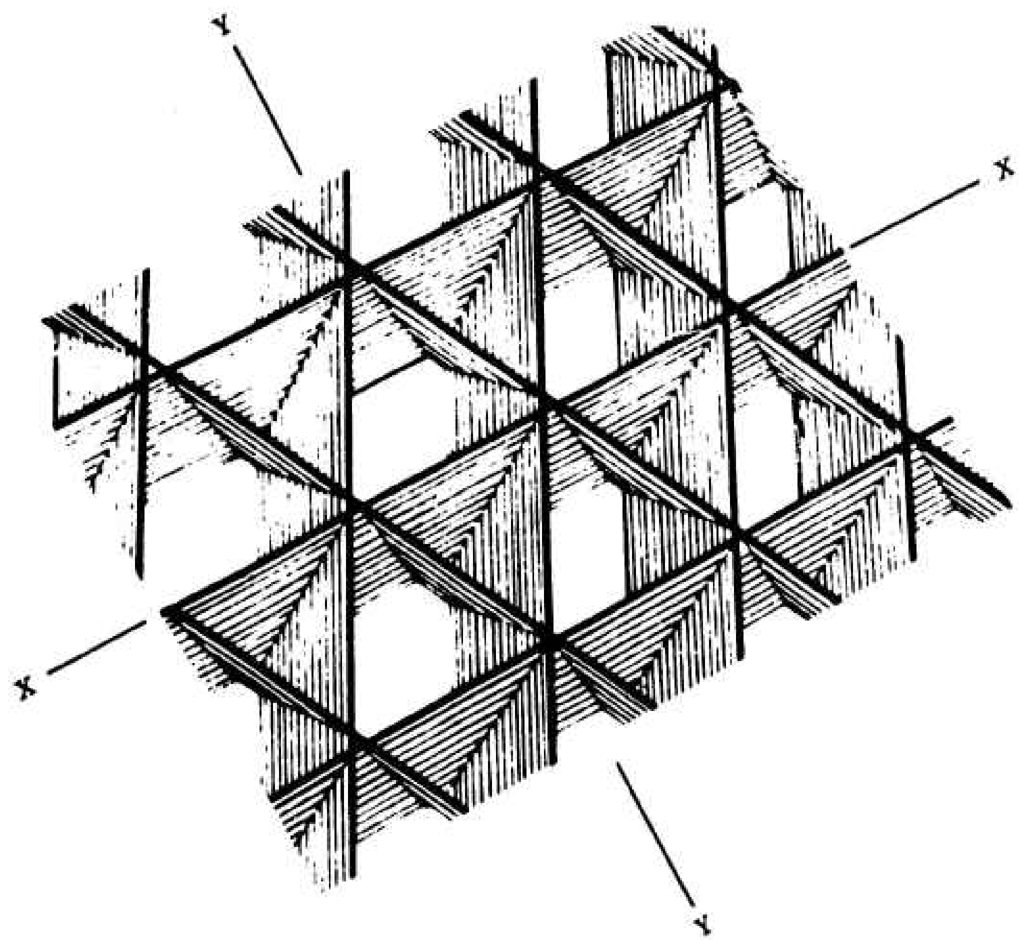


Figure 2. Basic Tetra-Core Element.

The finite-element method was used for this contract, along with an automatic grid generator to automatically idealize the structure and apply loads to it.

Two types of elements are used in this program: a linear strain triangle (6 nodes) for the legs of the Tetra-core, and a constant strain triangle (3 nodes) for the face sheets attached to the core. The linear strain triangle is required for the legs because of the change of strain from tension to compression between top and bottom of the tetrahedron when the panel is subjected to a bending moment. If a constant strain triangular element were used in this strain field, the result would be an averaging of the maximum and minimum stresses since the element would enforce a constant strain over the triangle. A constant strain triangle may be used for the face sheets, however, since the stress field does not contain changes of sign or large strain gradients. This allows a more efficient use of the available core storage space since additional nodes do not have to be generated at the midpoint of each side.

The program has been written to automatically generate finite-element models for Tetra-core structures made up of flat plates, cylinders, or airfoils. Two basic types of Tetra-core may be generated: a true Tetra-core, where the sides of the Tetra-core meet at a point to form a tetrahedron; and a truncated Tetra-core, in which the tops have been "cut off" of the tetrahedron. The basic tetrahedron geometry can be varied to give a skewed tetrahedron. Self-equilibrating nodal loads are automatically applied to the model based on the input combination of in-plane and out-of-plane loads. Multiple load cases may be run. Boundary conditions (nodal fixities) are applied to the model to hold it from translating or rotating without providing reactions for the nodal loads. This combination of loads and boundary conditions results in the idealization of a coupon of Tetra-core to which loads corresponding to those in some location of a larger structure are applied. A conventional finite-element analysis would be run of the total structure or a part of the structure. Stresses from this run would then be used to apply loads to the Tetra-core model corresponding to critical locations on the total structure.

Deflections are calculated using a Choleski triangularization method with substitution to solve the equation $[K]\ \{\delta\} = \{P\}$. Stresses and margin of safety are calculated at several locations in the linear strain elements. A buckling analysis of a triangle in each leg is done using the buckling equation of an orthotropic rectangle of the same size as a triangle in that leg, since no orthotropic triangle buckling equation are available. An optimization of the Tetra-core geometry can be made

using a steepest descent/side-step method to find a minimum weight design for the given loading conditions. A nonlinear analysis can be run using stress-strain curves of the material in each leg to account for the change in stiffness of each plate element with increasing load. The effect of a hole in a Tetra-core structure can be analyzed. The program will give a zero stiffness to plates connecting to a node at the center of the hole. This causes the loads to be redistributed into plates around the hole, creating the effect of a stress concentration.

The program is now configured for the IBM 360/44, but it can be easily modified to run on any 360. Converting to a CDC 6600 would be more difficult, since direct-access disc drives are required.

Figure 3 shows the elements of the program and the sequence of calculation. Each program element, input and output data, and model usage are described in the following chapters. This includes input format, output format, and a complete listing of the program.

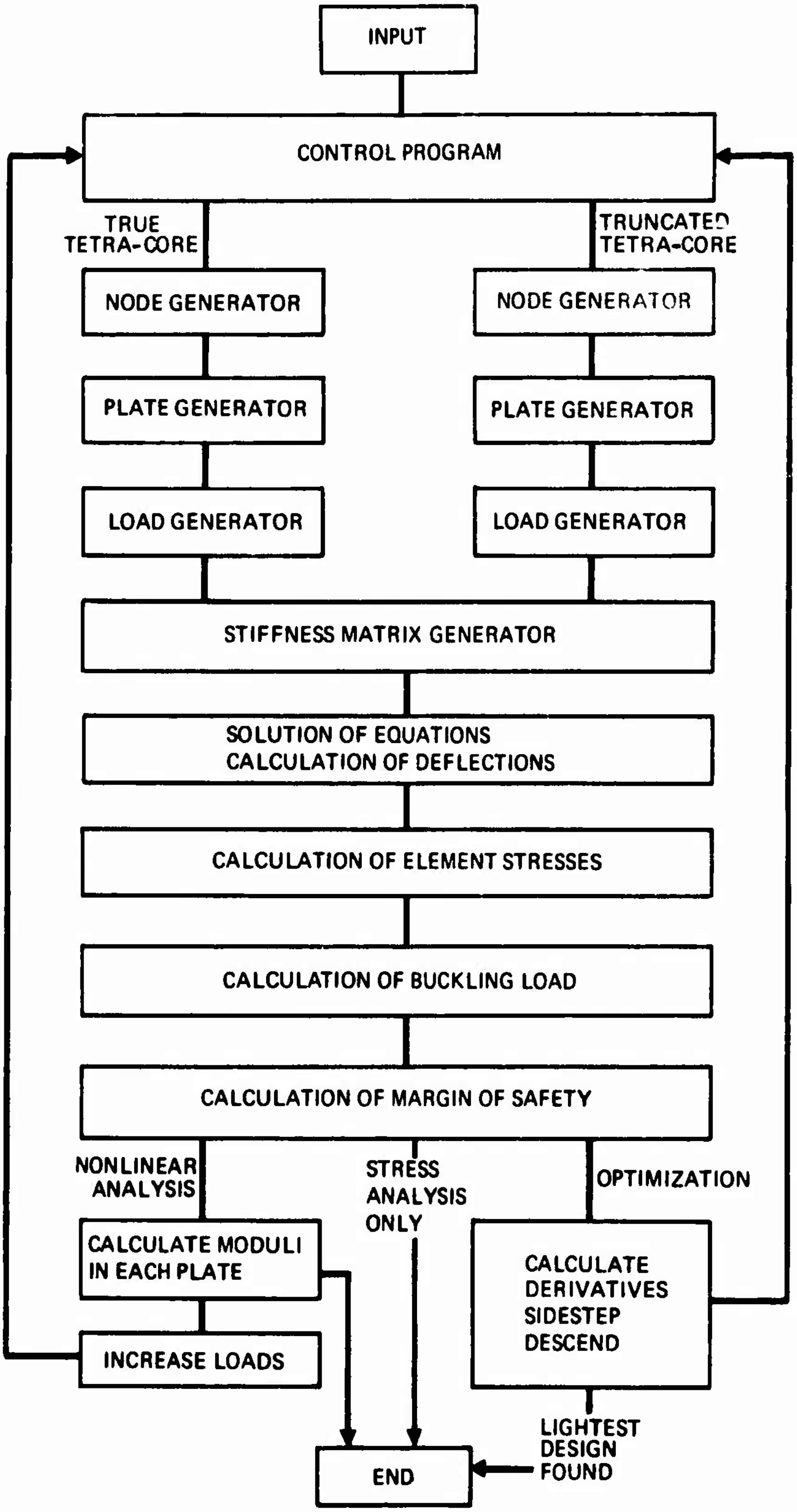


Figure 3. Sequence of Operations in Tetra-Core Analysis Program.

MODEL DESCRIPTION

This program has been written to automatically generate a finite-element model as a series of node points in space and connect them with plate elements into an idealization of a Tetra-core structure (Figure 1). It will automatically apply nodal loads to the model to represent uniform applied loads. Tetra-core flat plates, cylinders, and airfoils can be analyzed. The program uses the finite-element model it has generated to calculate stresses and deflections for each element and each node. A margin of safety is calculated for each plate element. A simple buckling analysis is performed on a plate in each leg. An optimization can be done to determine the least-weight Tetra-core design for a given set of loads. A nonlinear analysis can be performed by inputting a set of material stress-strain curves and applying loads in increments to compute the degradation in each plate element. The effect of a hole in the model can be simulated by giving a low stiffness to the plates connected to a specified node. A diagram of the sequence of computations performed by the program is given in Figure 3.

INPUTS

Input data required to run the Tetra-core analysis program has been kept to a minimum to simplify the work required for trade-off studies on the effect of geometry, materials, and other variables. Basic inputs are:

- Type of analysis to be run--nonlinear, optimization, damage, face sheets, flat plate, cylinder, airfoil, and true or truncated tetrahedron
- Optimization controls
- Size of specimen to be analyzed and tetrahedron geometry
- Material data--moduli and Poisson's ratio, allowable stresses, and stress-strain curves
- Loading to be imposed on the model

Using these data, the program generates a set of node points and plates to model the Tetra-core element and applies the specified loadings as nodal loads.

The basic tetrahedron geometry, shown in Figures 4 and 5, is input as the length of a side, the actual tetrahedron height, the theoretical height at which the apexes would meet, and the offsets which determine the amount of deviation from a true tetrahedron.

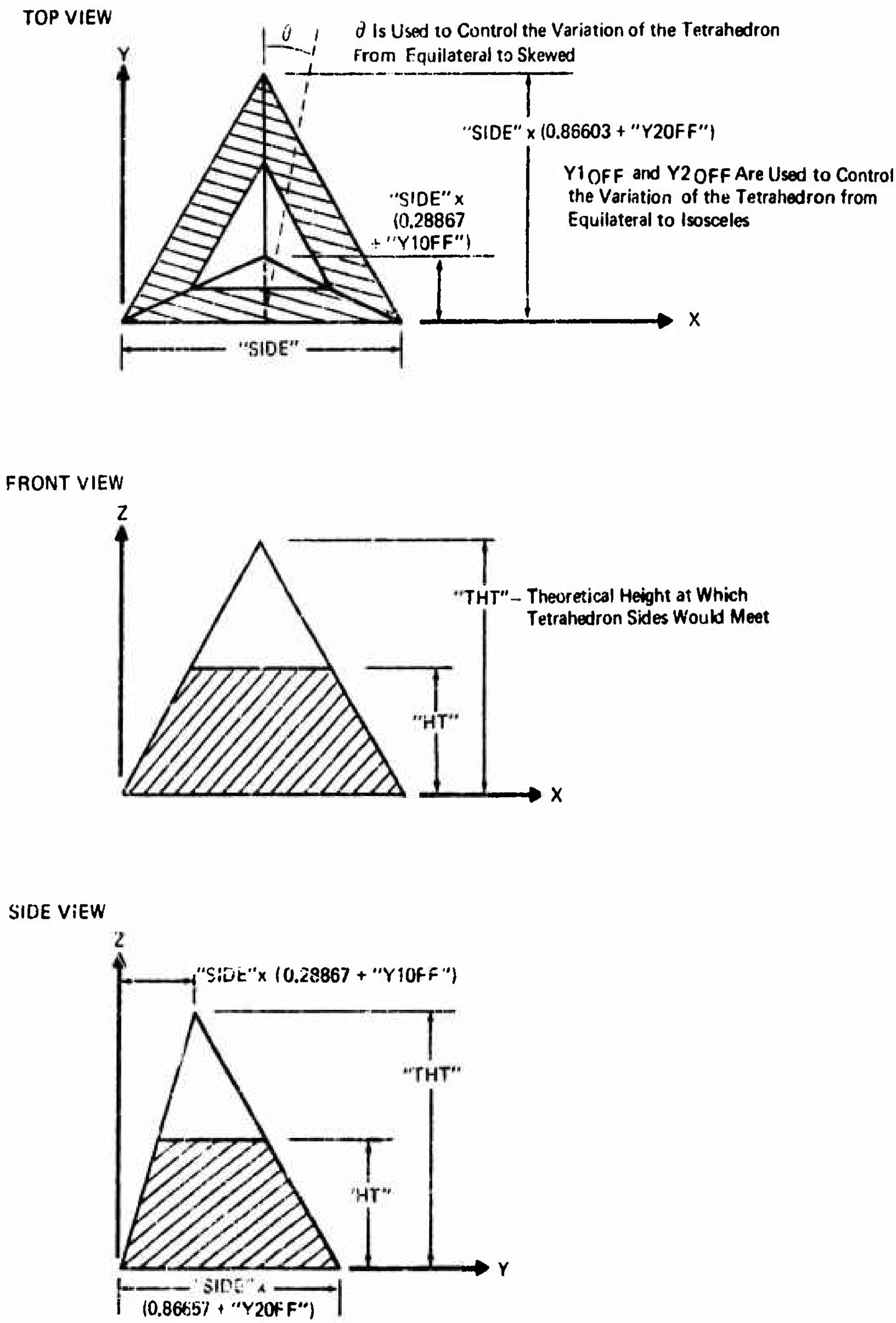


Figure 4. Basic Tetrahedron Geometry.

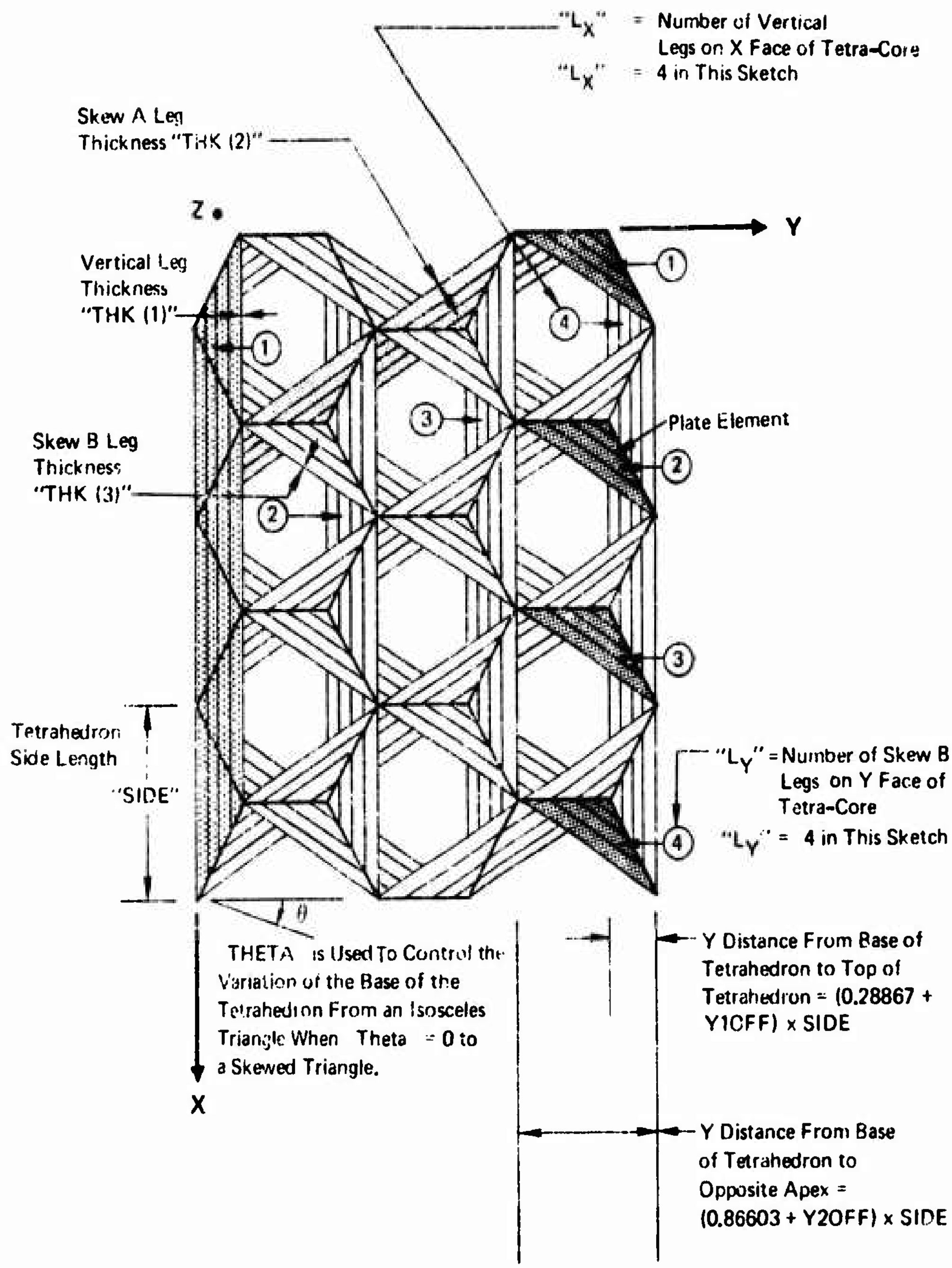


Figure 5. Tetra-Core Input Geometry.

If the actual height and the theoretical height are identical, then a true tetrahedron will result. Two generation options are available in the program: a true tetrahedron and a truncated tetrahedron. The true tetrahedron option should be used whenever possible, since it requires fewer nodal points and plates to be generated, and uses a more efficient nodal numbering system that allows a smaller bandwidth when used in a cylinder or an airfoil. If the theoretical height for a truncated tetrahedron is input as a large number (such as 10 inches), then a three-sided figure with near-vertical walls will result. The truncated Tetra-core model cannot be used to generate a true Tetra-core, since three node points at the top of a tetrahedron would have the same coordinates. This would cause a singular stiffness matrix and result in a bad solution. Offsets can be used to change the shape of the figure from a true tetrahedron with angles of 60 degrees, if no offsets are used, to an isosceles tetrahedron of any angle, if offset is used, to a skewed tetrahedron, if θ is used. The number of legs on the X and Y faces (LX and LY) must be even numbers (2, 4, 6, etc.).

Plates to connect the node points and form a Tetra-core model are generated so that the local X-axis is always in the direction of the fibers in that leg, as shown in Figure 6. For sandwich faces, the local X-axis is in the global X direction. Elastic moduli, allowable stresses, and stress-strain curves are input separately for each leg and each face sheet, thus allowing each leg to be a different material if desired. Material stress-strain curves are input only if a nonlinear analysis is run.

Using the input data LX and LY and the geometry, the tetrahedrons are automatically integrated into a Tetra-core node system as shown in Figure 7 for a true tetrahedron and in Figure 8 for a truncated tetrahedron. The node point numbering system is shown for each type. Other examples are shown in Appendix I.

The number of plates generated depends on the input number of legs. Plates are divided into Vertical, Skew A, and Skew B legs by the program. In Figures 9 and 10, the Skew A and B plates generated for a true tetrahedron with 10 legs on the X face and 6 legs on the Y face are shown. Plates are numbered in sequence with plates in the Vertical legs generated first, then plates in Skew A legs, then plates in Skew B legs. When cylinders and airfoils are generated, several additional plates are added to "zip up" the seam and connect the two sides with more Skew A and Skew B plates. Figure 11 shows the Skew A plates making up a cylinder. Face sheets can be automatically added to the true Tetra-core model to give a sandwich element. It was not considered feasible to generate a face sheet for the truncated Tetra-core element due to the complex shapes which would be required.

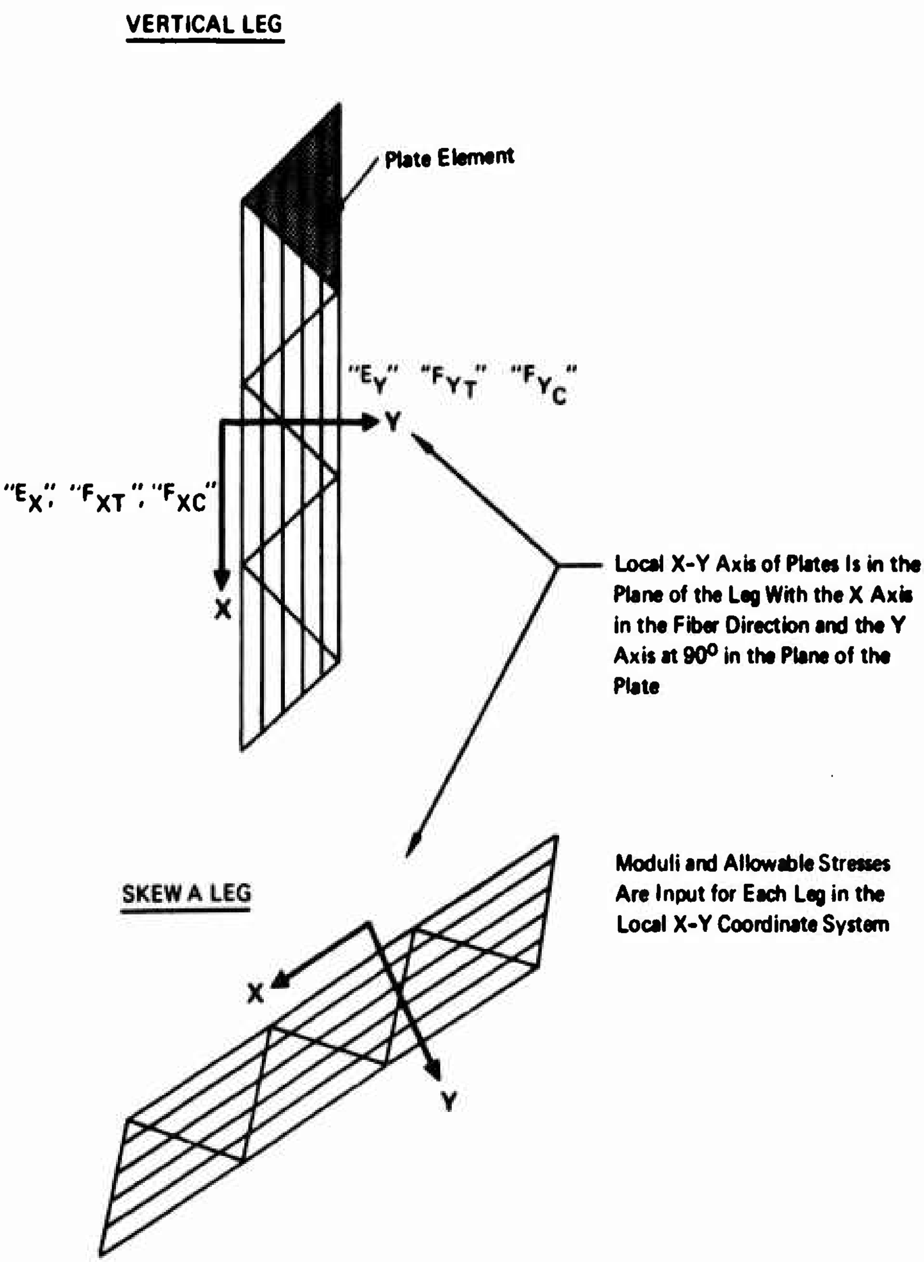


Figure 6. Material Properties Required as Input.

- Nodes Connected by Dark Lines Are on Upper Surface of Plate
- Nodes Connected by Light Lines Are on Lower Surface of Plate

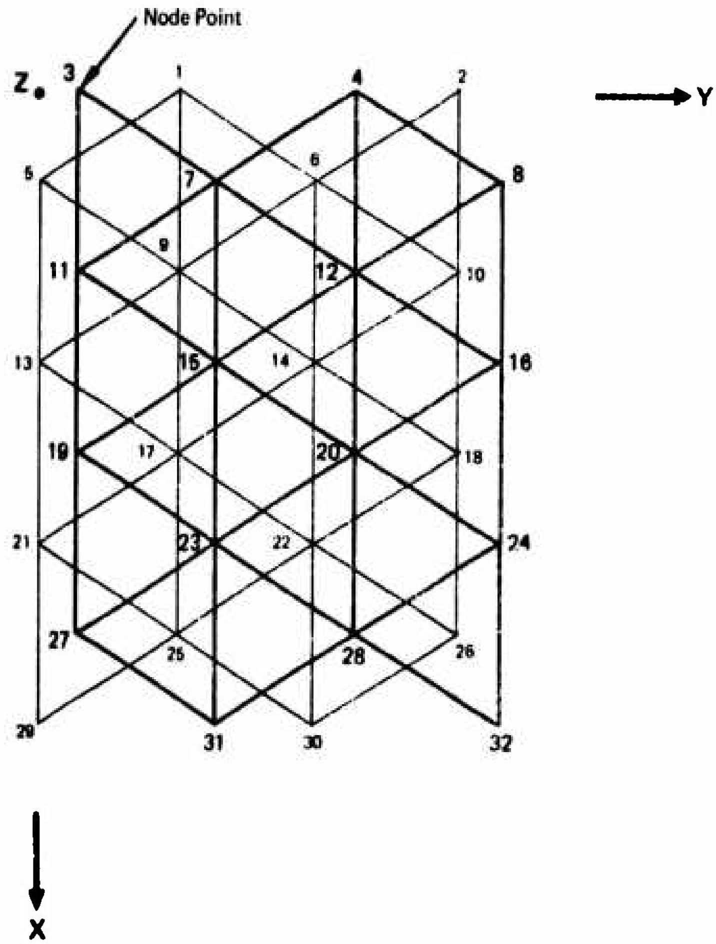


Figure 7. Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

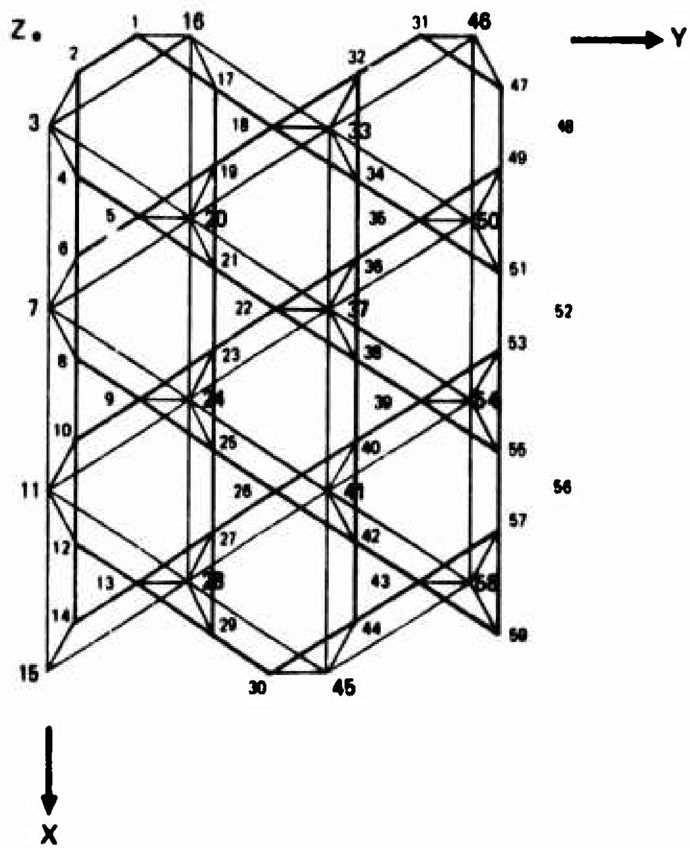


Figure 8. Nodal Numbering System, Truncated Tetra-Core Flat Plate, L_X = 4, L_Y = 4.

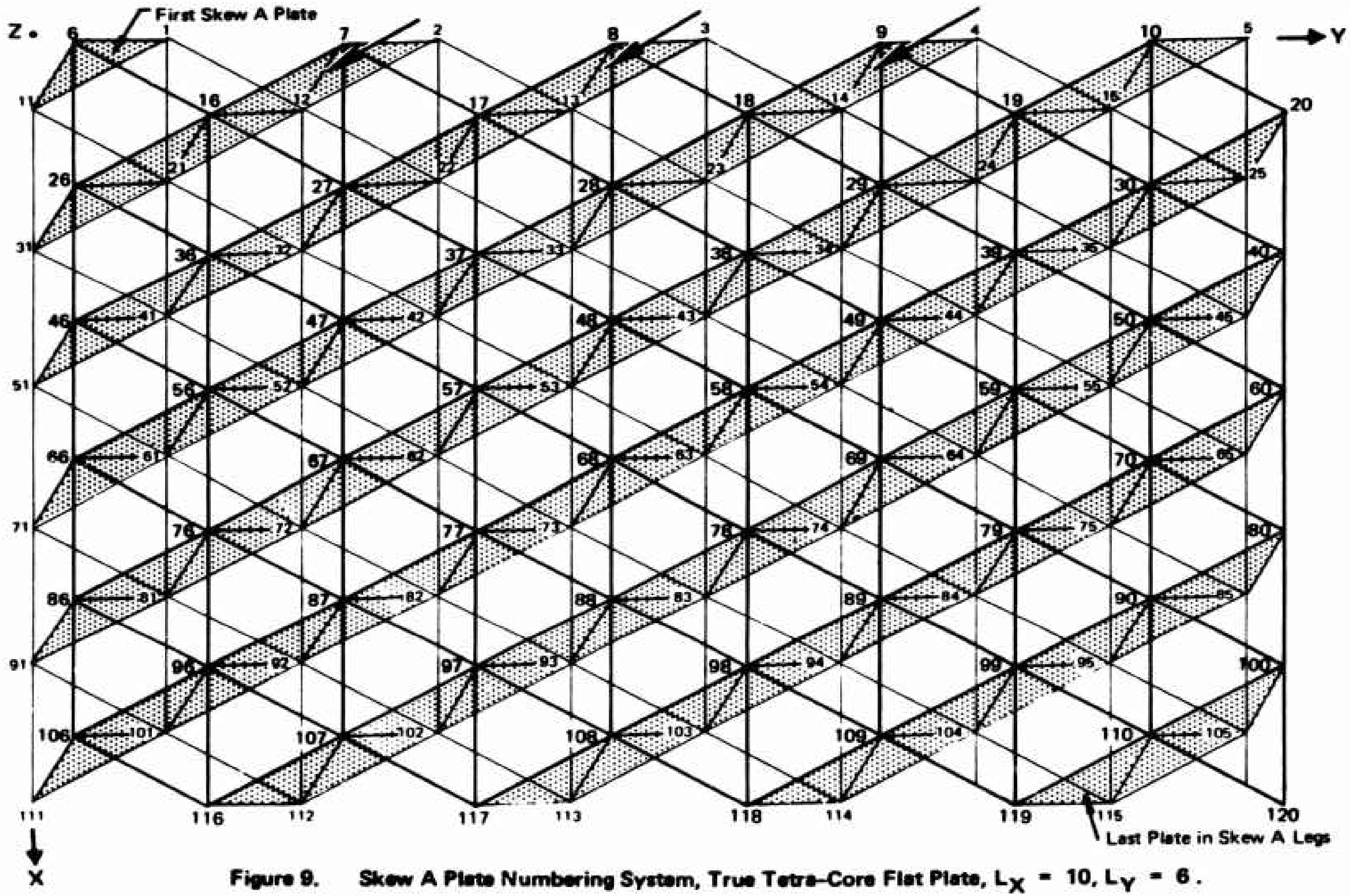


Figure 9. Skew A Plate Numbering System, True Tetra–Core Flat Plate, $L_X = 10, L_Y = 6$.

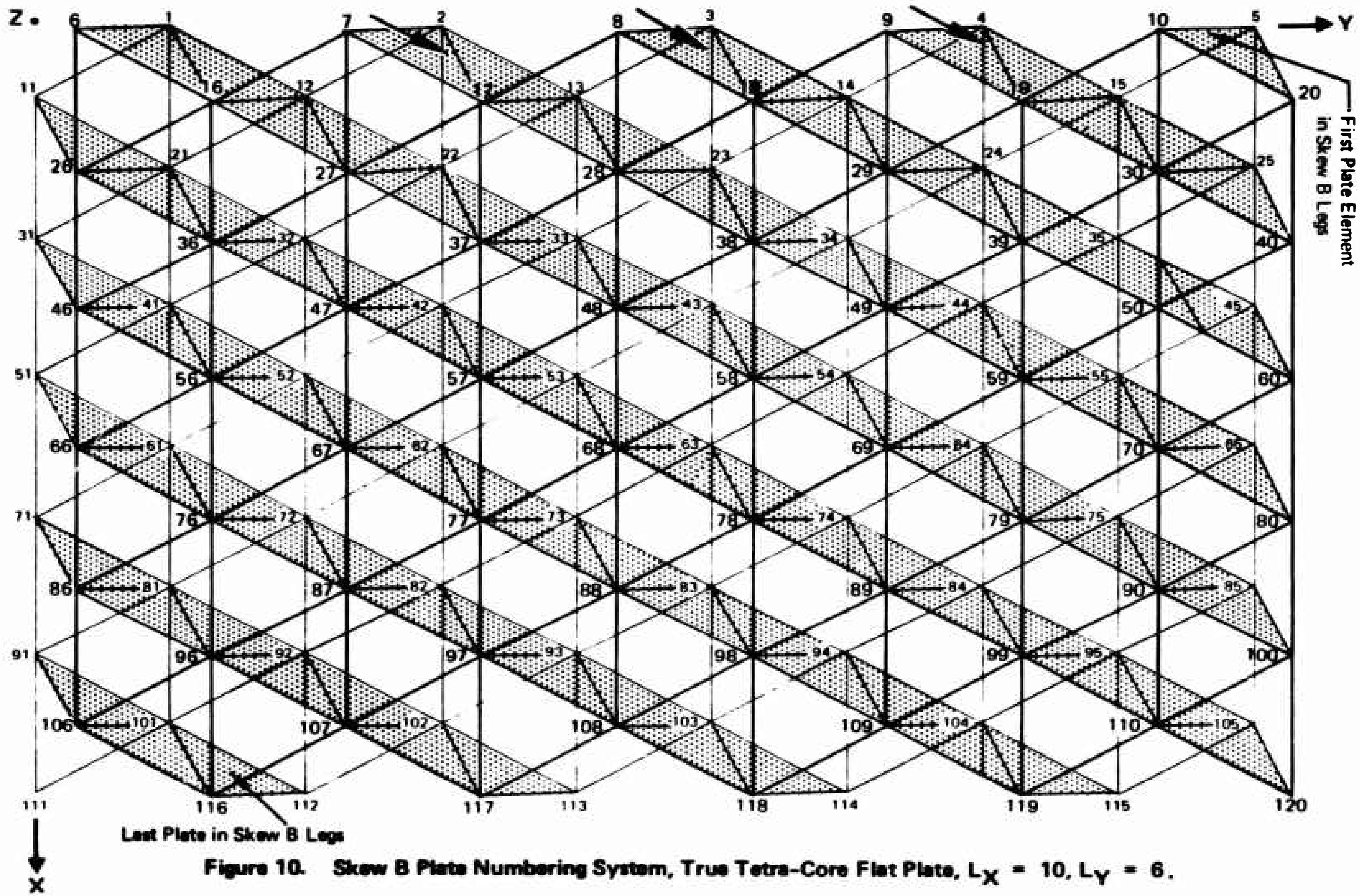


Figure 10. Skew B Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$.

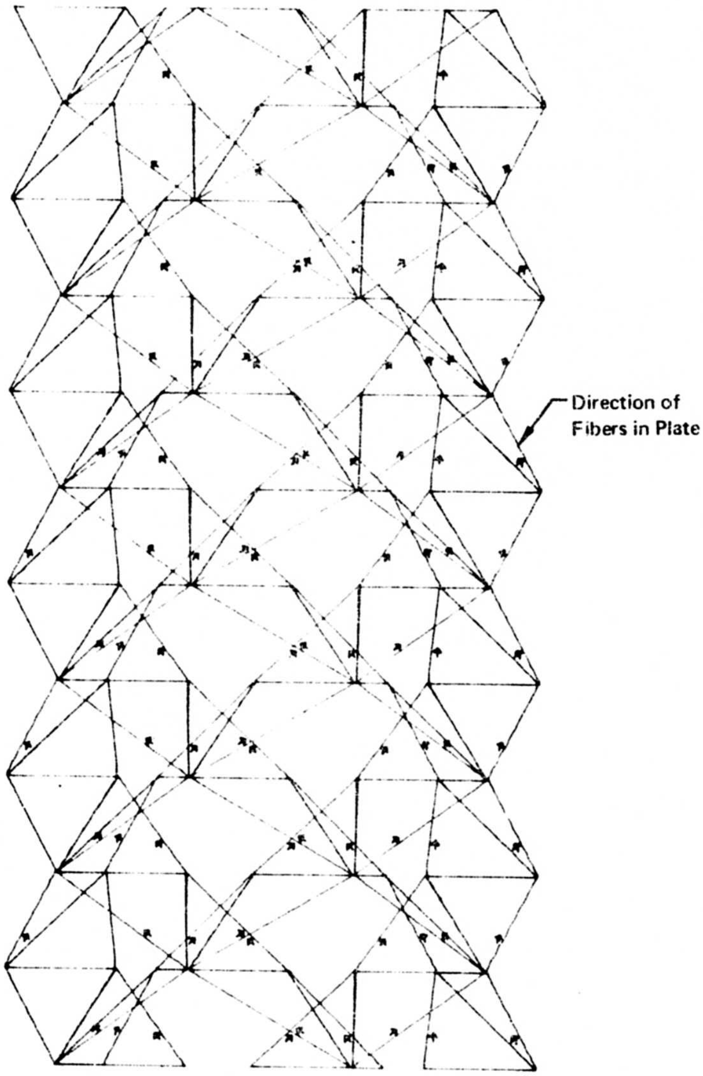


Figure 11. Skew A Plates in Cylinder (Computer Plot).

When generating a cylinder, the program transforms the flat plate Y and Z coordinates into a circular shape with a circumference equal to the Y dimension of the plate. Thus the outside radius of the cylinder is the X dimension of the plate divided by 2π. The airfoil section is generated with an input set of thickness/chord and X/chord ratios to define its shape. Vertical legs are defined to be parallel to the X-axis for cylinders and airfoils to give the maximum bending stiffness.

Node point loads are automatically applied to the finite-element model based on the input loading conditions. Loads are automatically applied in self-equilibrating sets; i.e., a load applied to one side is reacted by an equal and opposite load on the opposite side. Loading of the linear strain triangle, which is used for the Vertical, Skew A, and Skew B legs, includes a load on the mid-node point between the vertex node points. This load is four times the load at a vertex for a uniform applied load. The vertex node point loadings generated by inputting flat plate end loads in the X and Y directions and shear loads for the true tetrahedron case are shown in Figures 12a, b, and c. For moment loads, the load on the center node point of the triangle side is zero, since the two vertex loads cancel out. Node point loadings generated from input moments in the X, Y, and XY directions are shown in Figures 12d, e, and f for the true tetrahedron. Out-of-plane shear loads can be applied to the model. A moment is automatically applied to react the moment caused by the shear loads as shown in Figures 12g and h for vertical shear loads on the X and Y faces.

The truncated tetrahedron is loaded in a similar manner, except that the lower node points in an edge are given twice the load of the upper node points, since there are twice as many upper node points. The mid-side node point load is four times the upper node point load. More detailed examples of loads applied to Tetra-core flat plates are shown in Appendix I.

The cylinder model can be loaded with moment, torque, and end loads. The input load is used as the total load on the section and is divided among all nodes on that end. The nodal loads generated from an input torsion load on a cylinder are shown in Figure 13.

The airfoil model can be loaded with moment, torque, and end loads in a similar manner to the cylinder. A torsion load applied to a diamond airfoil section is shown in Figure 14.

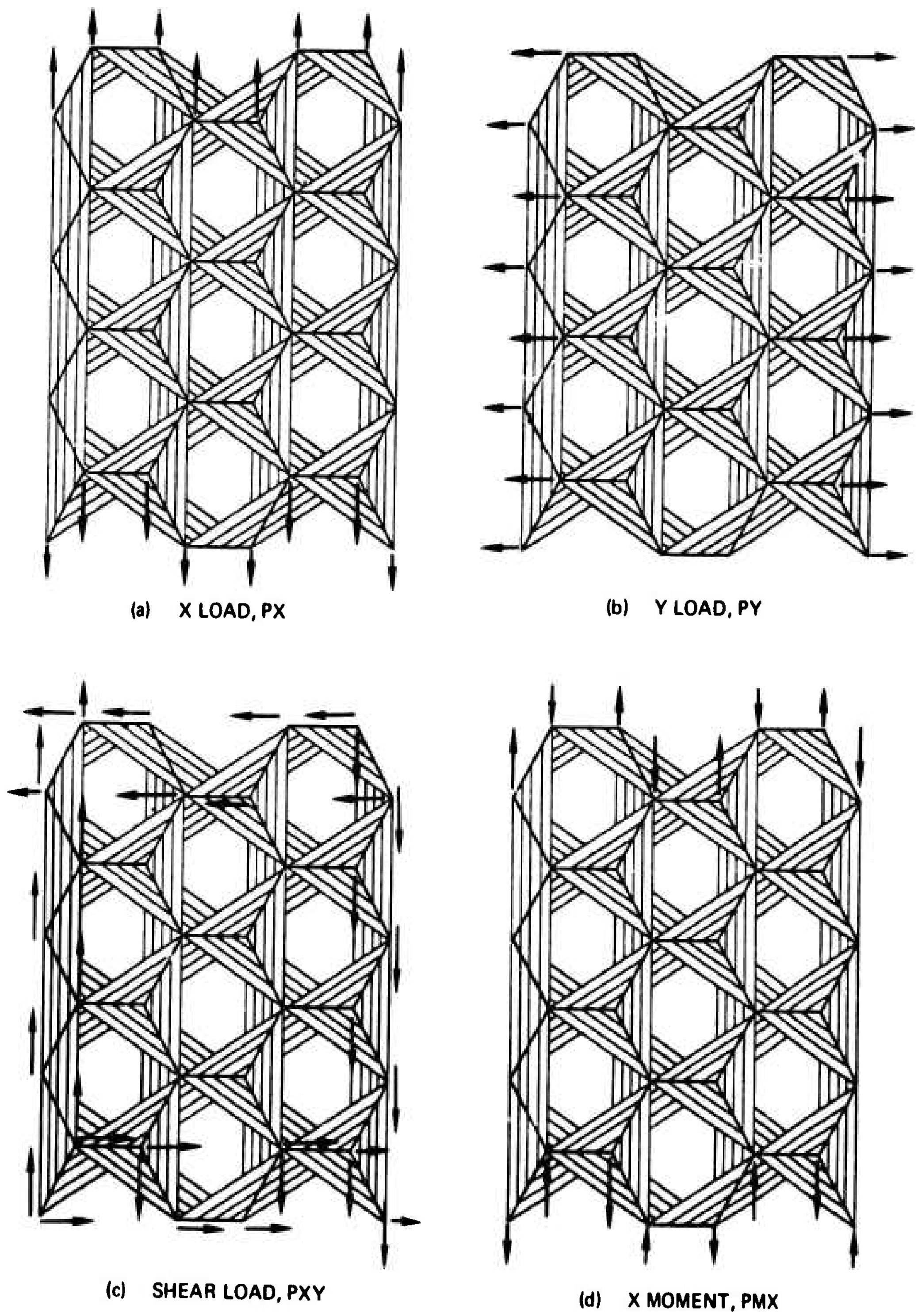
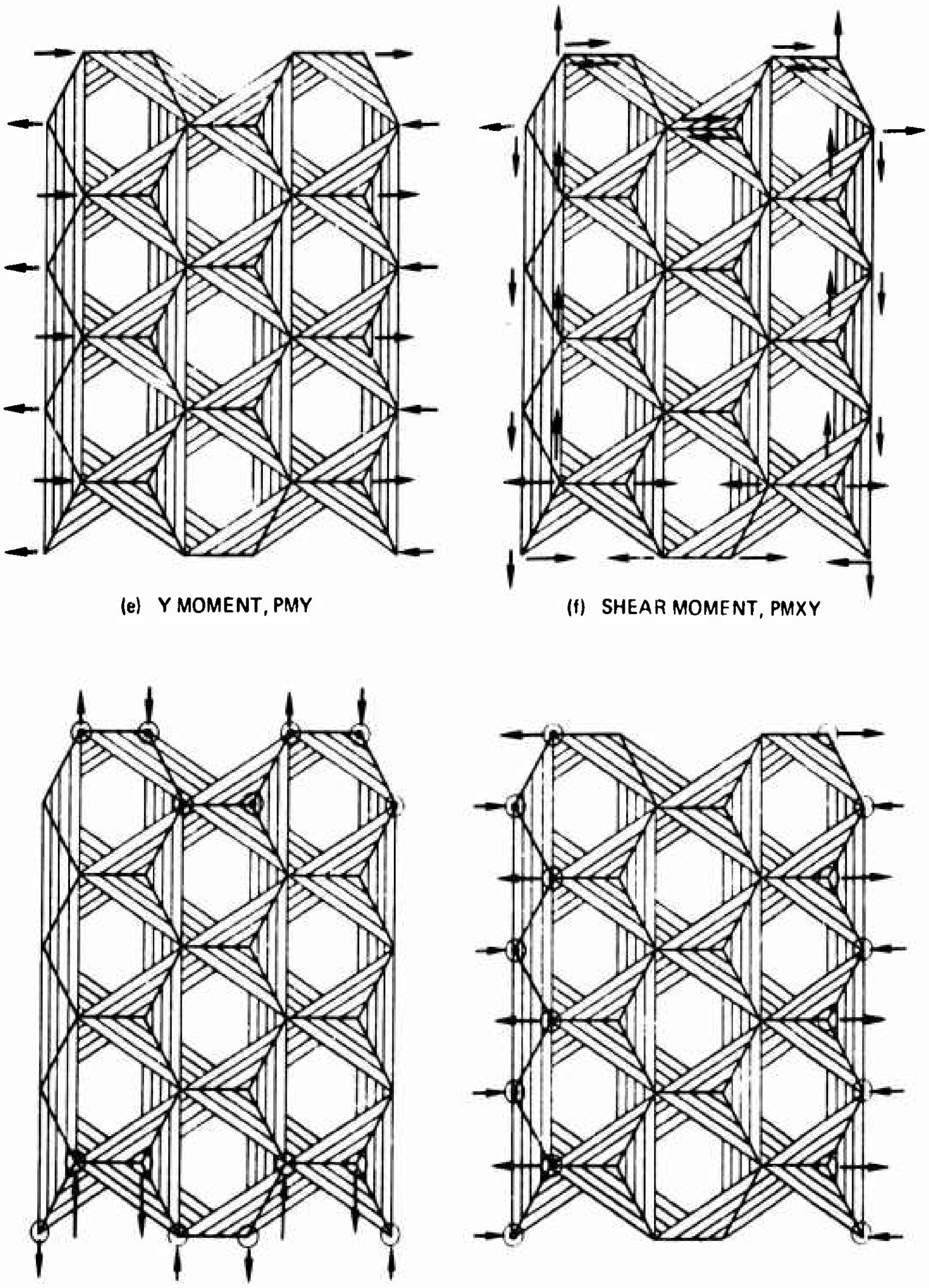


Figure 12. Flat Plate Applied Loads.



(e) Y MOMENT, PMY

(f) SHEAR MOMENT, PMXY

(g) X SHEAR, XQSHR

(h) Y SHEAR, YQSHR

Figure 12. Continued.

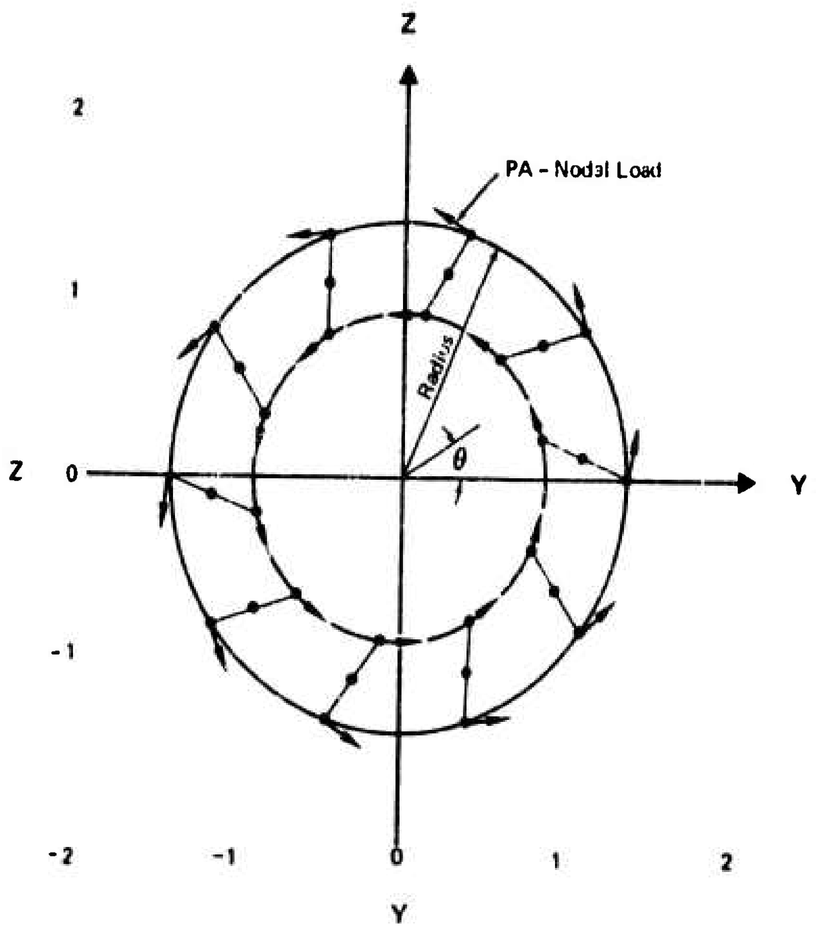


Figure 13. True Tetra-Core Cylinder With Torsion Load Applied, $L_X = 10$.

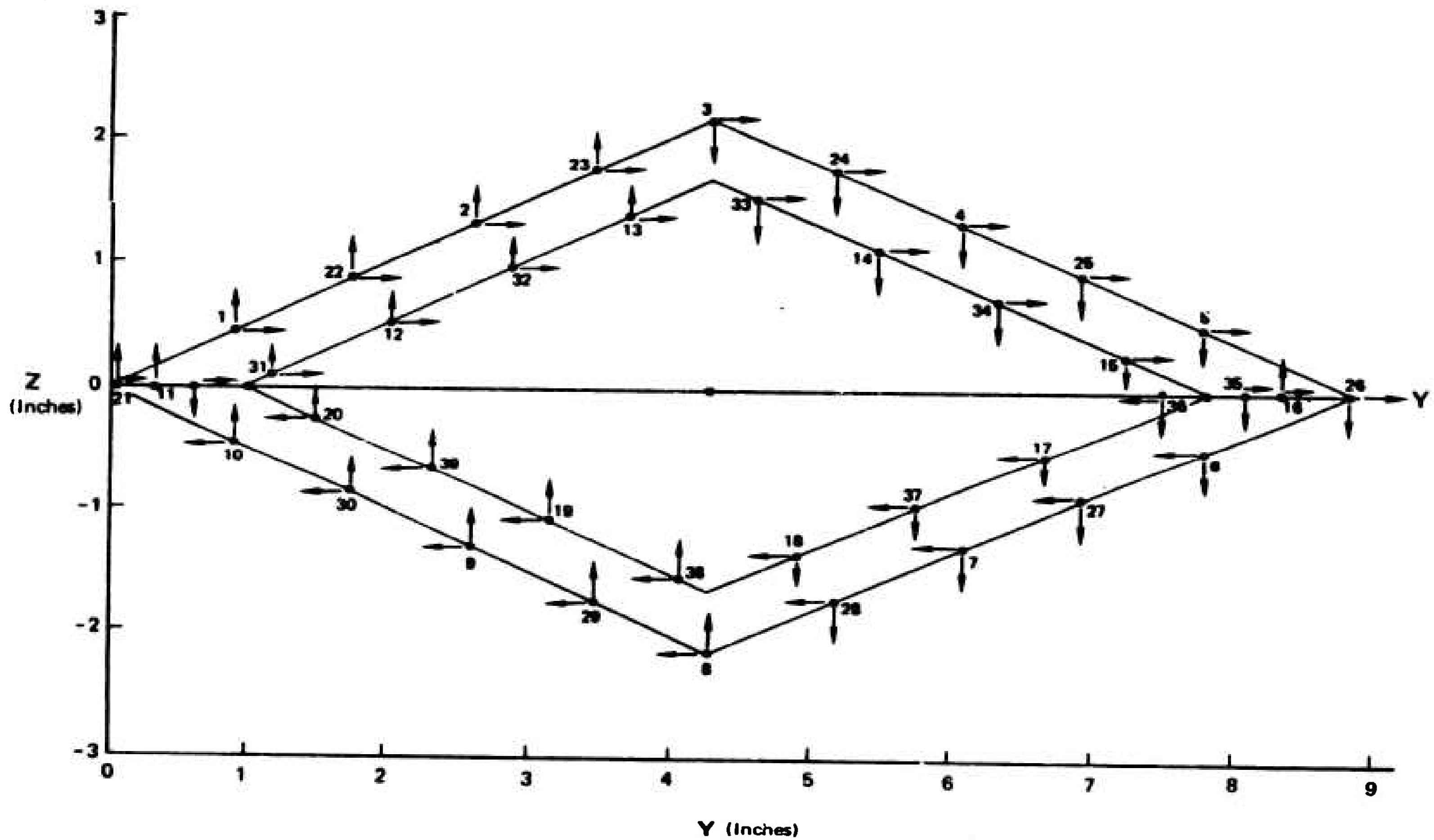


Figure 14. True Tetra-Core Airfoil With Torsion Load Applied, $L_X = 20.$

Loads can be applied to the true Tetra-core specimen being analyzed in any combination (end loads, moments, vertical shears, etc.). The truncated Tetra-core specimen can be loaded with end loads and moments in any combination, but if a vertical shear load is applied, then only certain combinations of loads may be applied at the same time, because of the fixities that are applied to the four sides of the panel to keep the model stable. If no vertical shear loads are applied, the Z-freedom is fixed for all nodes on the panel sides. If a vertical shear is applied on the X face, the side nodes are fixed in the Y direction, which permits movement in the Z direction. A vertical shear load on the Y face causes edge nodes to be fixed in the X direction. Thus, if a load in the Y direction is applied to a panel with a vertical shear load on the X face, the effect of the Y loads will be lost, since they are applied to nodes that cannot move in the direction of the load.

Moment, torsion, and end loads can be applied in any combination to both true and truncated Tetra-core cylinders and airfoils.

The effect of a hole in the Tetra-core can be simulated by giving the plates connected to a node in the center of the panel a zero stiffness. This unloads those plates and transfers the load to the remaining plates, providing the effect of a stress concentration around the hole.

AUTOMATIC MODEL GENERATOR

The Tetra-core analysis program has been written so that a finite-element model will be generated from the input data.

- Type of Analysis--Card Type 1
- Optimization Controls--Card Type 2
- Geometry--Card Types 5, 6, and 7
- Material Data--Card Types 4, 8, 9, and 10
- Loading--Card Types 12 and 13

Flat Plate Nodal Generator

A different nodal generator subroutine has been written for each type of Tetra-core, true and truncated. Subroutines NODGEN and NODGAN are used to generate nodal coordinates for the true and truncated Tetra-core models, respectively. True Tetra-core nodes are generated starting at the Y-axis, with numbers increasing as the X distance increases, as shown in Figure 7. Nodal coordinates (X, Y, Z) on the bottom surface of the Tetra-core at a given X distance are generated first,

using the repeating geometry of Tetra-core, then those on the upper surface. Then the X distance is increased and another cycle of numbering is begun. For the truncated Tetra-core, a different system is used because of its more difficult geometry. Numbering is started at the Y-axis on the first vertical leg, as shown in Figure 8. Node numbering follows the first vertical leg to its end, then switches to the second vertical leg, etc.

Nodal coordinates are generated in order of their appearance on the leg, resulting in a mixing of nodes from the upper and lower surfaces. Nodes in the last vertical leg are generated in the same manner, but every fourth node is unconnected to any other node in the flat plate and must be given fixity in the X, Y, Z directions. If a cylinder or airfoil is generated, these extra nodes are used to connect the two sides of the plate after it has been folded so that the two edges almost meet.

Cylinder Nodal Generator

Cylinders are generated from the flat plate coordinates previously generated. The cylinder is assumed to have a circumference equal to the length of the X face of the flat plate plus the width of one tetrahedron. This gives the cylinder an outside radius that is the circumference divided by 2π. The new Y and Z coordinates of each node are calculated:

$$\theta_i = y_i \text{ flat plate } {}^{*2\pi/\text{ circumference}}$$

$$y_{cyl} = \text{radius} * \cos(\theta_i)$$

$$z_{cyl} = \text{radius} * \sin(\theta_j)$$

The flat plate is essentially rolled into a cylinder so that the two edges of the flat plate are separated by the width of the one tetrahedron side in the Y direction. The same method is used for both true Tetra-core and truncated Tetra-core. An example of the end of a true Tetra-core cylinder is shown in Figure 13.

Airfoil Nodal Generator

Airfoil nodal coordinates are generated by transformation of flat plate coordinates using a curve-fitting subroutine. The same basic method is used for both true Tetra-core and truncated Tetra-core. The flat plate already generated is essentially folded over to make the upper and lower surfaces of the airfoil. The chord length of the airfoil is half the length of the X face of the original flat plate. The Z coordinate of each node is determined using the curve-fitting subroutine with the input t/C and X/C values. The distance

of a node from the X-axis and the airfoil chord length is used to calculate the X/chord ratio of that node. This X/chord ratio is used in the curve-fitting subroutine to determine the thickness/chord ratio of a point on the surface of the airfoil from the input t/C ratios of the airfoil. The Z coordinate of the node is then the thickness/chord ratio for that point times the chord length. See Figure 14 for an example of the end of a diamond-shaped airfoil generated by the program.

Plate Element Generator

Plate elements are generated automatically to connect the nodes from the nodal generator and form either a true Tetra-core or a truncated Tetra-core plate. True Tetra-core plate elements are generated in subroutine PLATGN, with truncated Tetra-core plate elements generated in subroutine PLATGA. The same method is used for both models, but a different subroutine is needed for each, since the nodal numbering sequence is different for each.

Flat Plate

Plate element generation starts with the upper plate in the left vertical leg. Linear strain triangular and quadrilateral plates are used for the Vertical and Skew A and B legs, since they can be expected to encounter compression strain on one side of the leg and tension strain on the other for some load cases. The mid-side node on the outer face of the element is reduced out, since it will not be connected to another element and would cause the merged stiffness matrix to be singular if left in. Two linear strain triangles are added to get the quadrilateral linear strain element used in the truncated Tetra-core model. See Figure 24 for the triangle and quadrilateral element node numbering sequence.

Plates are numbered so that the local X-axis of each plate element is in the fiber direction of that leg. Vertical plates are generated by following the vertical leg from top to bottom, then moving to the next vertical leg to the right, and starting at the top again. Each vertical leg has the same number of vertical elements.

Skew plate elements are then generated, starting with the plate in the top left of the flat plate segment, as shown in Figure 9. Skew A legs are generated from upper left to lower right. Generally, each Skew A leg has a different number of plate elements. Next, Skew B linear strain plate elements are generated, starting at the upper right corner of the segment, as shown in Figure 10. Leg generation proceeds from upper right to lower left, with a different number of plate elements in each leg.

If the face sheet option has been specified ("IFACE" = 1 on Card 1), an upper and lower face sheet will be generated for the segment.

Cylinder and Airfoil

A cylinder is generated by folding a flat plate model into a circular cylindrical segment. The two edges of the flat plate facing in the Y direction end up facing each other, separated by the width of a tetrahedron side. The result is a cylinder with a strip missing down the side. Plates are added to connect the two sides of the cylinder, resulting in a complete cylinder, shown in Figure 11. The plates making up a Tetra-core airfoil are generated in the same way, with the flat plate model being folded over into the desired shape and plate elements added to "zip up" the seam where the two ends meet.

Flat Plate Loads

The load generator takes input loads, which are uniform loads (given in either pounds/inch or inch-pounds/inch), and applies them as nodal loads to give an equivalent total load. The same method is used for both true Tetra-core panels and truncated Tetra-core panels. Since a different nodal numbering system was used for each type of Tetra-core, a different subroutine was used for each. Subroutine LODGEN generates loads for the true Tetra-core model, while loads for the truncated Tetra-core model are generated in subroutine LODGAN. When generating nodal loads for the linear strain triangle, the mid-side node must be loaded to give an equivalent uniform load across the element. Theoretically, the load should be divided so that one-sixth of the total load on that plate goes into each corner node and two-thirds goes into the mid-side node[1]. This is done automatically in the program.

In the case of a uniform load in the X direction, the input load (pounds/inch) is multiplied by the panel width and divided by the number of nodes on that face, taking account of the fact that some nodes have more load than others.

$$PX = \frac{XLOAD \times XLGT}{NXND}$$

where XLOAD = Input load on the X face (lb/in.)

XLGT = Length of X face

NXND = Number of nodes on X face

NXND = 2 x LX + (2 x LX - 1) x 4 for true Tetra-core model

This gives the X direction load to be applied to each node (PX). Loads are applied to the apex nodes only in the LODGEN and LODGAN subroutines since mid-side nodes are generated in the stress analysis overlay. Nodal loads applied to give the effect of a uniform load in the X direction are shown in Figure 12a for a true Tetra-core flat plate.

Figure 12b shows a load in the Y direction applied to the true Tetra-core panel. The nodal load PY is calculated:

$$PY = \frac{YLOAD \times YLGT}{NYND}$$

where YLOAD = input load on the Y face (lb/in.)

YLGT = Length of Y face (in.)

NYND = 2 x LY + (2 x LY -1) x 4 for a true Tetra-core model

LY = Number of legs on Y face

For a uniform in-plane shear load, the loads shown in Figure 12c are applied. The nodal loads are calculated in the same way as for a uniform load in the Y direction, except that a different nodal load (PXYA or PXYB) is used for the X and Y faces.

$$PXYA = \frac{XYLOD \times XLGT}{NXND}$$

where XYLOD = input in-plane shear load (lb/in.)

A moment load in the X direction is applied, as shown in Figure 12d, for a true Tetra-core flat plate. For a moment load, the mid-side nodes are not loaded. Nodal loads are calculated using the moment load per inch, panel width, panel depth, and number of nodes on the side being loaded.

$$PMX = \frac{XMOM \times XLGT}{HT \times LX}$$

where XMOM = Input moment on X face (in.-lb/in.)

A moment load is applied in the Y direction, as shown in Figure 12e. The nodal loads are calculated:

$$PMY = \frac{YMOM \times YLGT}{HT \times LY}$$

where YMOM = Input moment on Y face (in.-lb/in.)

A twisting shear moment is applied as shown in Figure 12f. Nodal loads PMXYA and PMXYB are calculated separately for the X and Y faces of the panel:

$$PMXYA = \frac{XYMOM \times YLGT}{HT \times LY}$$

$$PMXYB = \frac{XYMOM \times XLGT}{HT \times LX}$$

where XYMOM = Input twisting moment (in.-lb/in.)

Out-of-plane shear loads can be applied to flat panels using the input loads XQSHR and YQSHR. A combination of a vertical shear load and a moment load is applied to each loaded end of the panel, as shown in Figures 12g and 12h. The moment load PMX is applied to balance the moment caused by the two vertical shear loads PZ. The resulting set of loads is self equilibrating, so no reactions will result at the fixed nodes.

Calculations of nodal loads for a given X and Y vertical shear are:

$$PZ = \frac{XQSHR \times XLGT}{NXND}$$

$$PMY = \frac{XQSHR \times SIDE \times (LY - 1) \times XLGT}{2 \times HT \times LX}$$

where XQSHR = Input vertical shear load on X face (lb/in.)

$$PZ = \frac{YQSHR \times YLGT}{NYND}$$

$$PMX = \frac{YQSHR \times SIDE \times (0.86603 + Y2\ OFF) \times (LX-1) \times YLGT}{2 \times HT \times LY}$$

where YQSHR = Input vertical shear load on Y face (lb/in.)

For a true Tetra-core flat plate, these vertical shear loads can be applied with any other loads, such as moment loads and in-plane shear loads; etc. for the truncated Tetra-core flat panel, all loads cannot be applied simultaneously. A vertical shear load YQSHR can be applied with loads and moments on the X face, but cannot be used with in-plane shears or twisting moments or with a moment on the Y face. Similarly, the vertical shear load YQSHR can be applied with loads and moments on the Y face, but not with shear loads or moment loads on the X face. The reason is that the truncated Tetra-core model is

unstable on the edges and must be supported to prevent rigid body rotation in some of the plates. This is done by fixing nodes on the flat plate edges in the Z direction. This allows the flat plate to deflect in the X-Y plane but not in the Z direction. With this fixity, no vertical shear load can be applied to the model because nodes that would be loaded in the Z direction are fixed in that direction and cannot be loaded. When a vertical shear load on the X face is specified, the program will fix the edge nodes in the Y direction, instead of the Z direction, which allows Z nodal loads to be applied. However, loads can no longer be applied to these nodes in the Y direction. Therefore, shear loads and moments on the Y face cannot be applied at the same time that vertical shear loads are applied to the X face. Similarly, if a vertical shear load is applied to the Y face, all edge nodes are fixed in the X direction, and shears and moments on the X face cannot be applied. For the same reason, a twisting shear moment cannot be applied in conjunction with an out-of-plane shear load.

Cylinder Loads Generation

Total end loads, bending moments, and torsion loads are applied to cylinders as nodal loads. End loads are applied to the cylinder by dividing the input X load (in pounds) by the number of nodes on one end of the cylinder, taking into account the fact that some nodes have a larger load than others.

$$PX = \frac{XLOAD}{NXND}$$

where XLOAD = input X total end load on cylinder (lb)

A torsion load is applied to the cylinder as nodal loads tangent to the surface of the cylinder at the node being loaded, as shown in Figure 13. Loads on the inner surface of the cylinder have a smaller load than those on the outer surface by the ratio of their radii.

$$PA = \frac{TORQ}{LX \times (r_A^2/r_A) \times 4}$$

$$PB = \frac{PA\ r_B}{r_A}$$

where TORQ = Input total torsion load on cylinder (in.-lb)

r_A = Radius of outer surface of cylinder

r_B = Radius of inner surface of cylinder

A bending moment is applied to the cylinder as a tension load on nodes below the Y-axis of the cylinder and a compression load on nodes above the Y-axis of the cylinder. The magnitude of the load on each node depends on the distance of that node from the Y-axis. That is, the neutral axis of the cylinder is assumed to be the Y-axis, and a straight-line variation of load with distance from the neutral axis is used.

$$PMX = \frac{XMOM}{SUM}$$

where XMOM = Input total moment on end of cylinder (in.-lb)

$$SUM = 4\sum\sum_{j=1}^{nodes} (Y_i)^2$$

Airfoil Loads Generation

End loads and bending moments are applied to the airfoil in the same way they were applied to the cylinder. A torsion load is applied to the airfoil by applying a load to each end node, in the direction of the airfoil surface at that node, as shown in Figure 14. Since no prediction could be made of the location of the shear center of the airfoil, it was necessary to apply the same total load to each node. This was done instead of ratioing the magnitude of each load by its distance from the shear center, as was done for the cylinder.

Nodal Fixity

In a finite-element analysis, it is necessary to restrain the model from translating or rotating as a rigid body, as this will result in a singular stiffness matrix. This is done by removing the ability of some nodes to move in the X, Y, or Z direction, thus "fixing" them in that direction.

For the Tetra-core flat plate analysis, nodes on the Y face next to the X-axis (nodes 3, 5, 11, 13,..., 29 in Figure 7) are fixed in the Y direction. The first two nodes on the first vertical leg (nodes 3 and 5 in Figure 7) are fixed in the X direction. The lower of the two nodes is fixed in the Z direction. This keeps the model from translating or rotating as a rigid body.

The boundary conditions and system of applied loads used for the flat plate Tetra-core model result in a coupon type analysis. That is, the plate segment being analyzed can be thought of as a small coupon cut out of the larger structure, with loads corresponding to the loads in the larger structure at that point applied to the coupon sides. This allows the

use of a coarse-grid finite-element analysis, using a standard finite-element program, to determine the loads at any point in the structure. Tetra-core analyses are then run of several points in the structure, using loads found from the coarse-grid analysis.

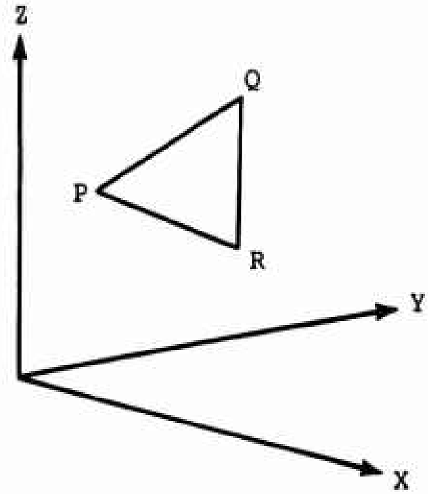
For the truncated Tetra-core model, additional fixities are applied to make the model stable, as discussed in the previous section. Nodes on the panel edge are given fixity in the Z direction to restrain rigid-body rotation of some edge plates. This can be illustrated by looking at the Skew B leg in the upper right corner of a panel (Figure 8). It can be seen that this plate element can pivot about its lower surface node (No. 46) without inducing any in-plane strains into the Skew A or vertical plates supporting it. This results in a singular stiffness matrix, but can be prevented by restraining movement of the edge nodes in the X, Y, or Z direction. If a vertical shear load is applied on the X faces of the panel, then edge nodes are given a fixity in the Y direction instead of the Z direction. If a shear load is applied on the Y faces, then edge nodes are given a fixity in the X direction, as discussed in the Load section.

For Tetra-core cylinders and airfoils, nodes on the end next to the X-axis are given fixity in the X, Y, and Z directions, providing the effect of a cylinder with one end fixed into a stiff foundation.

FINITE-ELEMENT STRESS ANALYSIS

Linear Strain Element

To best derive the stiffness matrix for a linear strain, six-node triangular element, we first require the geometry expressed in terms of some local in-plane coordinate system.



If $X_{QP} = X(Q) - X(P)$, etc.,

$Y_{QP} = Y(Q) - Y(P)$

$Z_{QP} = Z(Q) - Z(P)$

and $\ell_1 = \sqrt{\left[X_{QP}^2 + Y_{QP}^2 + Z_{QP}^2\right]}$

where ℓ_1 is the length of the line PQ.

Then the direction cosines of PQ are

$\lambda_1 = X_{QP}/\ell_1$ for $\ell_1 > 0$

$\lambda_2 = Y_{QP}/\ell_1$

$\lambda_3 = Z_{QP}/\ell_1$

or $\lambda_1 = \lambda_2 = \lambda_3$ for $\ell_1 \leq 0$

The direction cosines of the normal to PQ, through R are obtained as follows:

with $RR = \lambda_1 X_{RP} + \lambda_2 Y_{RP} + \lambda_3 Z_{RP}$

$$X_1 = X_{RP} - \lambda_1 RR$$

$$Y_1 = Y_{RP} - \lambda_2 RR$$

$$Z_1 = Z_{PP} - \lambda_3 RR$$

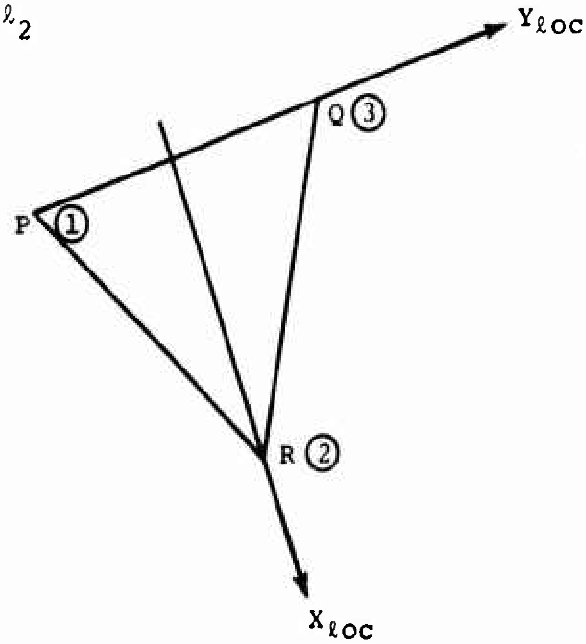
and $\ell_2 = \sqrt{\left[X_1^2 + Y^2 + Z^2\right]}$

and the direction cosines of the normal are

$$\mu_1 = X_1/\ell_2$$

$$\mu_2 = Y_1/\ell_2$$

$$\mu_3 = Z_1/\ell_2$$



Referring the triangle to local coordinates shown previously and denoting $X_{loc_i} - X_{loc_j}$ by X_{i_j}, we have

$$a_3 = X_{21} = X_{QP}\lambda_1 + Y_{QP}\lambda_2 + Z_{QP}\lambda_3$$

$$-b_3 = Y_{21} = X_{QP}\mu_1 + Y_{QP}\mu_2 + Z_{QP}\mu_3$$

$$a_1 = X_{32} = X_{RQ}\lambda_1 + Y_{RQ}\lambda_2 + Z_{RQ}\lambda_3$$

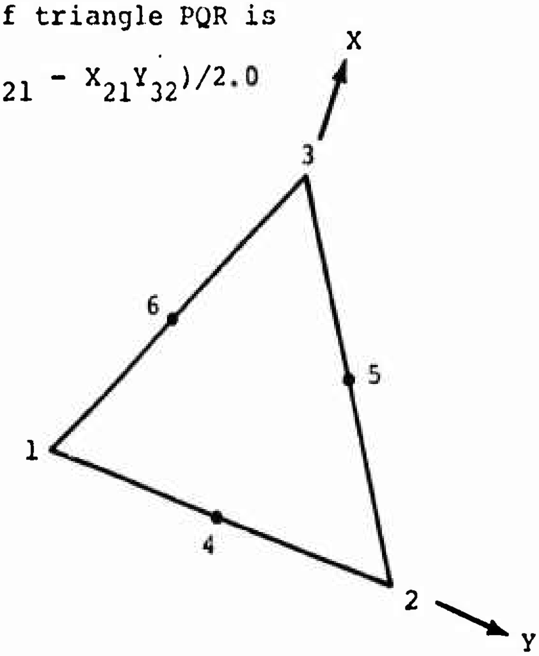
$$-b_1 = Y_{32} = X_{RQ}\mu_1 + Y_{RQ}\mu_2 + Z_{RQ}\mu_3$$

$$-a_2 = X_{31} = X_{RP}\lambda_1 + Y_{RP}\lambda_2 + Z_{RP}\lambda_3$$

$$b_2 = Y_{31} = X_{RP}\mu_1 + Y_{RP}\mu_2 + Z_{RP}\mu_3$$

and the area of triangle PQR is

$$\Delta = (X_{32}Y_{21} - X_{21}Y_{32})/2.0$$



Following the method of Felippa,[2] the stiffness matrix of the six-node linear strain triangle may be expressed as

$$\left[B^T\right]\begin{bmatrix} C_{11}Q & C_{12}Q & C_{13}Q \\ C_{21}Q & C_{22}Q & C_{23}Q \\ C_{31}Q & C_{32}Q & C_{33}Q \end{bmatrix}\left[B\right]$$

where B relates the strain at any point within the triangle to the nodal deflections,

and $Q = \frac{At}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$ (3 X 3)

and $\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_{12} \end{Bmatrix} \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} = \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \alpha_{12} \end{Bmatrix}$

Using Hooke's Law to relate stress and strain, $C_{23} = C_{32} = C_{31} = C_{13} = 0$, and further, C_{ij} is symmetric.

Matrix B may be expressed as follows:

with $\phi_X = \frac{1}{2\Delta} \begin{bmatrix} 3b_1 & -b_2 & -b_3 & 4b_2 & 0 & 4b_3 \\ -b_1 & 3b_2 & -b_3 & 4b_1 & 4b_3 & 0 \\ -b_1 & -b_2 & 3b_3 & 0 & 4b_2 & 4b_1 \end{bmatrix}$

(3 X 6)

and $\phi_Y = \frac{1}{2\Delta} \begin{bmatrix} 3a_1 & -a_2 & -a_3 & 4a_2 & 0 & 4a_3 \\ -a_1 & 3a_2 & -a_3 & 4a_1 & 4a_3 & 0 \\ -a_1 & -a_2 & 3a_3 & 0 & 4a_2 & 4a_1 \end{bmatrix}$

(3 X 6)

then $B = \begin{bmatrix} \phi_X & 0 \\ 0 & \phi_Y \\ \phi_Y & \phi_X \\ \multicolumn{2}{c}{(9 \times 12)} \end{bmatrix}$

So

$$k = \begin{bmatrix} \phi_x^T & 0 & \phi_y^T \\ & & \\ 0 & \phi_y^T & \phi_x^T \end{bmatrix} \begin{bmatrix} C_{11}Q & C_{12}Q & 0 \\ C_{12}Q & C_{22}Q & 0 \\ 0 & 0 & C_{33}Q \end{bmatrix} \begin{bmatrix} \phi_x & 0 \\ 0 & \phi_y \\ \phi_y & \phi_x \end{bmatrix}$$

which may be expanded to

$$k = \left[\begin{array}{c|c} \phi_x^T C_{11} Q \phi_x + \phi_y^T C_{33} Q \phi_y & \phi_x^T C_{12} Q \phi_y + \phi_y^T Q C_{33} \phi_x \\ (6 \times 6) & (6 \times 6) \\ \hline \phi_y^T C_{22} Q \phi_x + \phi_x^T C_{33} Q \phi_y & \phi_y^T C_{22} Q \phi_y + \phi_x C33 Q \phi_x \\ (6 \times 6) & (6 \times 6) \end{array}\right]$$

When k is partitioned as above, a small increase in efficiency of computing results as

1/ Multiplication by null matrices is avoided

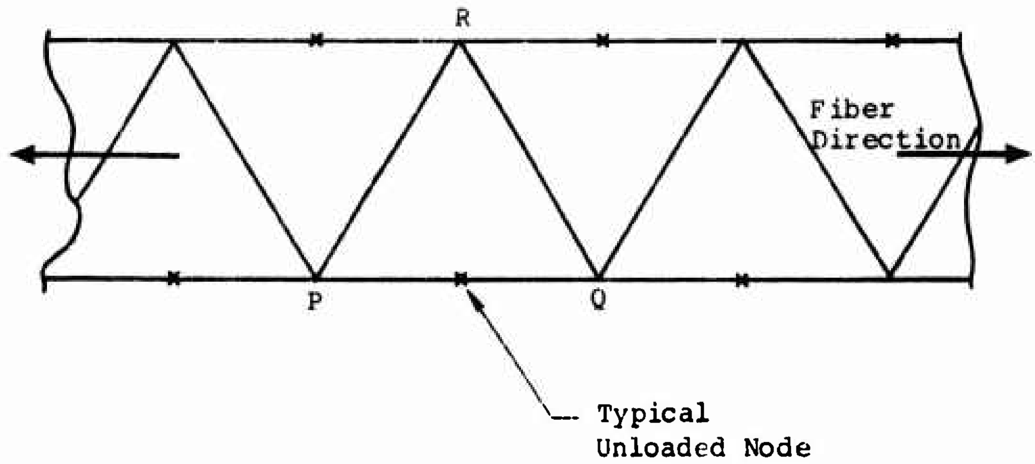
2/ Symmetry of the stiffness matrix can be used

3/ Storage for a full B matrix is not required.

With

$$\begin{Bmatrix} P_{X_1} \\ P_{X_2} \\ P_{X_3} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ P_{Y_5} \\ P_{Y_6} \end{Bmatrix} = [k] \begin{Bmatrix} U_1 \\ U_2 \\ U_3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ V_5 \\ V_6 \end{Bmatrix}$$

In the Tetra-core model, if P-Q is the fiber direction, then the node at the mid-side of P-Q is never loaded or supported out-of-plane.



Upon generating the structural stiffness matrix, this node would, in general, create linear dependence in the stiffness matrix. The linear dependence could be removed by giving the triangle bending stiffness, but this would greatly reduce program efficiency. As the node is never loaded, the simplest and most direct solution is to remove the node at the elemental stage.

This is done as follows: The node to be removed corresponds to node 6. So, when k is rearranged so that it corresponds to the usual order, the rows and columns are rearranged so that those corresponding to U_6, V_6 occur at the bottom and right-hand side of the matrix.

$$\left\{\begin{array}{c} P_{X_1}\\ P_{Y_2}\\ P_{X_2}\\ P_{Y_2}\\ P_{X_3}\\ P_{Y_3}\\ P_{X_4}\\ P_{Y_4}\\ P_{X_5}\\ P_{Y_5}\\ \hline P_{X_6}\\ P_{Y_6} \end{array}\right\} = \left[\begin{array}{c|c} K_{11} & K_{12}\\ \hline K_{21} & K_{22} \end{array}\right] \left\{\begin{array}{c} U_1\\ V_1\\ U_2\\ V_2\\ U_3\\ V_3\\ U_4\\ V_4\\ U_5\\ V_5\\ \hline U_6\\ V_6 \end{array}\right\}$$

and the matrix is partitioned so that

$$\left\{\begin{array}{c} P_{1-5}\\ \hline P_6 \end{array}\right\} = \left[\begin{array}{c|c} K_{11} & K_{12}\\ \hline K_{21} & K_{22} \end{array}\right] \left\{\begin{array}{c} U_{1-5}\\ \hline U_6 \end{array}\right\}$$

P_{X_6}, P_{Y_6} are each zero, so

$$\{P_{1-5}\} = [K_{11}]\ [U_{1-5}] + [K_{12}]\ [U_6]$$

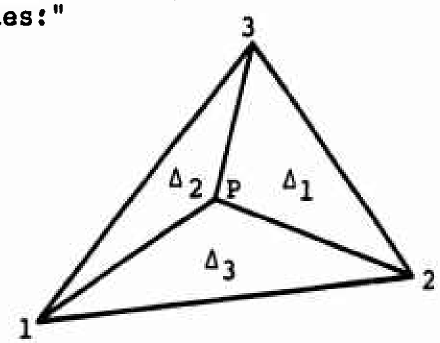
$$\{0\} = [K_{21}]\ [U_{1-5}] + [K_{22}]\ [U_6]$$

i.e., $\{P_{15}\} = [K_{11}] - [K_{12}][K_{22}]^{-1}[K_{21}] \quad U_{1-5}$

and we have the stiffness of the element expressed in terms of the five remaining nodes as

$$[K_{11}] - [K_{12}]^T[K_{22}][K_{21}] = K_{RED}$$

In calculating the stress matrix, as in Reference 1, we define "area coordinates:"



$$\Delta = \Delta_1 + \Delta_2 + \Delta_3$$

$$\xi_1 = \frac{\Delta 1}{\Delta};\ \xi_2 = \frac{\Delta 2}{\Delta};\ \xi_3 = \frac{\Delta 3}{\Delta}$$

Coordinates ξ_i uniquely determine location P, and the stress matrix is

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{21} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix} \begin{bmatrix} \xi_i & 0 & 0 \\ 0 & \xi_i & 0 \\ 0 & 0 & \xi_i \end{bmatrix} \begin{bmatrix} B \end{bmatrix} \begin{Bmatrix} U_1 \\ V_1 \\ U_2 \\ V_2 \\ \cdot \\ \cdot \\ U_6 \\ V_6 \end{Bmatrix} i = 1 \rightarrow 3$$

In the program this has been broken down into a more efficient form as

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix} \begin{bmatrix} \underset{1X6}{\langle \psi_X \rangle} & 0 \\ 0 & \underset{1X6}{\langle \psi_Y \rangle} \\ \underset{1X6}{\langle \psi_X \rangle} & \langle \psi_X \rangle \\ \multicolumn{2}{c}{3 \text{ X } 12} \end{bmatrix} \begin{Bmatrix} U_1 \\ V_1 \\ \vdots \\ U_6 \\ V_6 \end{Bmatrix}$$

where

$$\underset{1X6}{<\psi_X>} = \frac{1}{2A} <(4\xi_1-1)b_1,\ (4\xi_2-1)b_2, (4\xi_3-1)b_3, 4(\xi_2 b_1+ {}_1b_2),$$

$$4(\xi_3 b_2+\xi_2 b_3, 4(\xi_1 b_3+\xi_3 b_1)>$$

$$\underset{1X6}{<\psi_Y>} = \frac{1}{2A} <(4\xi_1-1)a_1, (4\xi_2-1)a_2,\ (4\xi_3-1)a_3,\ 4(\xi_2 a_1+\xi_1 a_2),$$

$$4(\xi_3 a_2+\xi_2 a_3),\ 4(\xi_1 a_3+\xi_3 a_1)>$$

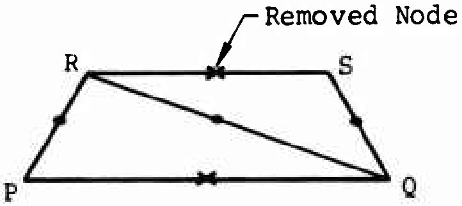
Multiplication by the C matrix is carried out after element strains have been determined in subroutine DEFL. The sixth node is reduced out of the stress matrix in the same manner as for the stiffness matrix.

$$[S\ T_{RED}] = [S\ T] - [S\ T_{12}]^T\ [K_{22}]^{-1}[K_{21}]$$

The stress matrix is determined for four locations in the linear strain triangle: Nodes P, Q, and R and the triangle centroid.

Quadrilateral Element

The linear strain quadrilateral element used in the truncated Tetra-core model is formed by adding two linear strain triangles from which the sixth nodes have been removed.



The mid-side node between R and Q is then reduced out as the top and bottom mid-side nodes had been, since it is never loaded or attached to another element. This results in a six-node linear strain quadrilateral. The stress matrices are similarly formed, with stress now being computed at the four corner nodes.

Constant Strain Triangle

The constant strain triangle used for face sheets is generated using the method of Przemieniecki[3]. The local in-plane coordinate system set up for the linear straing triangle

with $a_1 = X_{32}$, etc., is used. A [B] matrix relating strains within the element to deflections of the node points is defined.

$$[B] = \frac{1}{2A} \begin{bmatrix} b_1 & 0 & b_3 & 0 & b_2 & 0 \\ 0 & a_1 & 0 & a_3 & 0 & a_2 \\ a_1 & b_1 & a_3 & b_3 & a_2 & b_2 \end{bmatrix}$$

The stiffness matrix [K] is then

$$[K] = \int_V [B]^T \, [C] \, [B] \, dV$$

$$[K] = \Delta t \, [B]^T \, [C] \, [B]$$

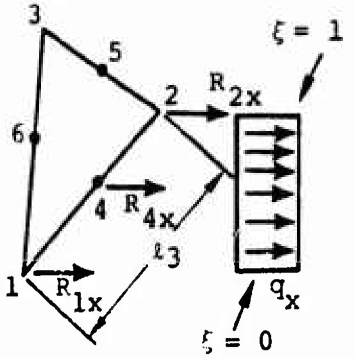
and

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \begin{bmatrix} C_{11} & C_{12} & 0 \\ C_{12} & C_{22} & 0 \\ 0 & 0 & C_{33} \end{bmatrix} [B] \begin{Bmatrix} U_1 \\ V_1 \\ U_2 \\ V_2 \\ U_3 \\ V_3 \end{Bmatrix}$$

Consistent Surface Loads Vector

The uniform in-plane loading acting on an edge of the linear strain triangle must be resolved into stress resultants acting at the two corner nodes and the midpoint. The derivation shown here is from a document by Tocher.[1] The total force on the plate is

$$P_X = q_X \ell_3$$



The equilibrium equations can be written using virtual work as

$$<\delta U_1 \;\; \delta U_4 \;\; \delta U_2> \begin{Bmatrix} R_{1X} \\ R_{4X} \\ R_{2X} \end{Bmatrix} = \int_0^1 \delta U(\xi)\; P_X \ell_3 d\xi$$

The virtual deflection $\delta U(\xi)$ can be obtained as (Ref. 1)

$$\delta U = <\delta U_1 \;\; \delta U_4 \;\; \delta U_2> \begin{Bmatrix} 1-3\xi + 2\xi^2 \\ 4\xi - 4\xi^2 \\ 2\xi^2 - \xi \end{Bmatrix}$$

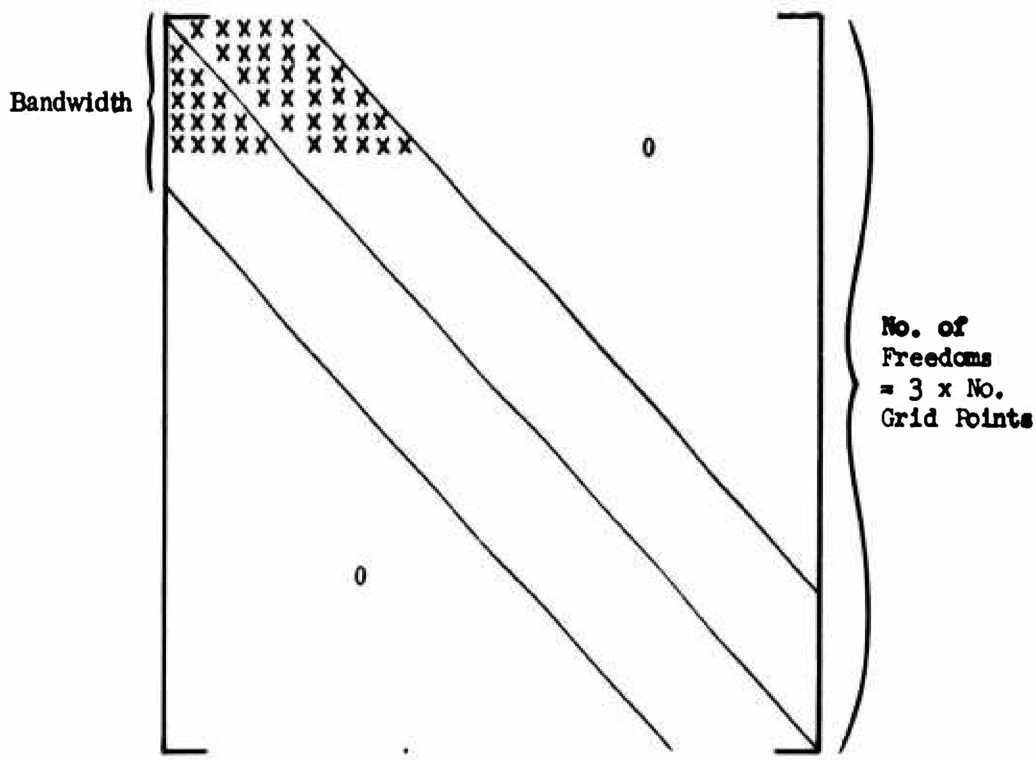
Thus

$$\begin{Bmatrix} R_{X_1} \\ R_{X_4} \\ R_{X_2} \end{Bmatrix} = q_X \ell_3 \int_0^1 \begin{Bmatrix} 1-3\xi + 2\xi^2 \\ 4\xi - 4\xi^2 \\ 2\xi^2 - \xi \end{Bmatrix} d\xi = \frac{q\ell_3}{6} \begin{Bmatrix} 1 \\ 4 \\ 1 \end{Bmatrix}$$

and the load distribution is simply two-thirds of the load through the middle node and one-sixth through each end node.

Calculation of Deflections

A solution of the equation $k\Delta = P$ is used to calculate deflections (Δ) from the stiffness matrix (k) and the loads vector (P). A banded form of the stiffness matrix is assumed, in which only the nonzero portion of the stiffness matrix, which is inside the bandwidth, is stored (Figure 15). The stiffness matrix is stored in blocks on the disc, with each block being three rows. Thus, only one block is in core at any one time during generation of the stiffness matrix. Each elemental stiffness matrix is generated and added into the overall stiffness matrix using the direct stiffness method. The location of the block which each member of the elemental stiffness matrix is added into is determined from the nodal connectivity of that element. Each element will add its stiffness to several blocks, thus requiring the ability to turn the needed block by means of a direct-access disc.



Core Storage Requirement = Bandwidth x No. of Freedoms

Figure 15. Structural Stiffness Matrix.

Once the stiffness matrix has been formed, it is reduced to a triangular form using the Choleski methods[3]. This results in all members of the matrix below the diagonal being zero.

$$\bar{k}_{ii} = \left(k_{ii} - \sum_{r=1}^{i-1} \bar{k}_{ir}^{2} \right)^{1/2} \qquad i = 1,n$$

$$\bar{k}_{ij} = \left(k_{ij} - \sum_{r=1}^{j-1} \bar{k}_{ir}\bar{k}_{rj} \right) / \bar{k}_{jj} \qquad i < j$$

$$\bar{k}_{ij} = 0 \qquad i > j$$

where n is the order of the stiffness matrix. The loads vector is modified by the same factors that were used to triangularize the stiffness matrix. With the triangularized form of the stiffness matrix, the deflections of the last node can be calculated

$$\bar{k}_{1n}\Delta_1 + \bar{k}_{2n}\Delta_2 + \cdots + \bar{k}_{nn}\Delta_n = P_n$$

where $\bar{k}$ is the triangularized stiffness matrix.

Since $\bar{k}_{1n}, \bar{k}_{2n} \ldots \bar{k}_{n-1,n} = 0$

$$\Delta_n = P_n/\bar{k}_{nn}$$

Using this method, $\Delta_{n-1}, \Delta_{n-2}, \ldots \Delta_1$ can be solved, in turn, by back substituting the previously found deflections into the calculations for the next higher row. For example, Δ_{n-1} can be calculated

$$\bar{k}_{n-1,n-1}\Delta_{n-1} + \bar{k}_{n,n-1}\Delta_n = P_{n-1}$$

where k_n had been calculated in the step before and

$\bar{k}_{n-1,n-1}$, $\bar{k}_{n,n-1}$ and P_{n-1} are known

This method of calculating deflections has proved to be much faster than the traditional method of inverting the stiffness matrix and multiplying $k^{-1}P$ to get Δ. The method allows a large problem to be solved on a computer with small core space,

since only three rows and three columns of the stiffness matrix are in core at one time during solution. However, it requires many accesses to the disc during a solution, since rows and columns must be shuffled in and out of core many times. On the 360/44 at Fort Eustis, the disc space available is the limiting factor on the size of problem that can be run.

Fixities are applied to the stiffness matrix by multiplying the diagonal term of the row to be fixed by a factor of 10^{12}. This has the effect of giving that freedom a much greater stiffness than other freedoms, effectively fixing it. This is equivalent to putting a one on the diagonal and setting elements of that row and column to zero, but it is more efficient.

OPTIMIZATION

A modified form of the steepest-descent optimization method is used to find a minimum weight Tetra-core design for a given set of loads[4]. Variables that can be optimized are tetrahedron geometry (side length, height, theoretical height, Y1 offset, Y2 offset, THETA) and the thicknesses of each leg. See Figures 4 and 5 for a description of the tetrahedron geometry. Optimization controls have been set up so that any combination of variables can be held constant during optimization. Also, any three variables can be forced to be the same during optimization; i.e., Vertical and Skew A and B legs can be kept to the same thickness during optimization. Input Card Type 2 is used to specify which variables will be optimized.

The optimization method will be illustrated using a two-dimensional example so that the variation of design variables can be plotted graphically. For the example, assume that a Tetra-core run is being made in which all variables are held constant except for the vertical leg thickness and the skew leg thickness, and that the Skew A and B legs will have the same thicknesses. Using this model, a design space can be plotted for a given set of loads in which the X and Y axes correspond to the variation in vertical and skew leg thicknesses as shown in Figure 16. For any point in the design space, the Tetra-core element will have a given weight and margin of safety. A series of curves of constant weight and another series of curves of constant margin of safety can be drawn as shown in the figure. The margin-of-safety curve for a margin of safety of zero is used to divide the design space into a feasible region, in which the margin of safety is greater than zero, and an unfeasible region, with a margin of safety less than zero. The optimization program does not know the location or shape of these curves, except for the location of the present design, but it can calculate the gradient of these curves; that is, the direction of travel in design space which will result in the maximum change in either the weight or the margin of safety. The gradient can be calculated by changing each variable, by a small amount, while holding all other variables constant and calculating the weight and margin of safety for the new design point. This is done separately for all variables, resulting in a set of partial derivatives:

$$\frac{\partial MS}{\partial t_i} = \frac{MS_i - MS_{BASE}}{t_i - t_{iBASE}} \qquad i = 1, N, \; N = \text{No. of variables}$$

$$\frac{\partial WT}{\partial t_i} = \frac{WT_i - WT_{BASE}}{t_i - t_{iBASE}} \qquad i = 1, N$$

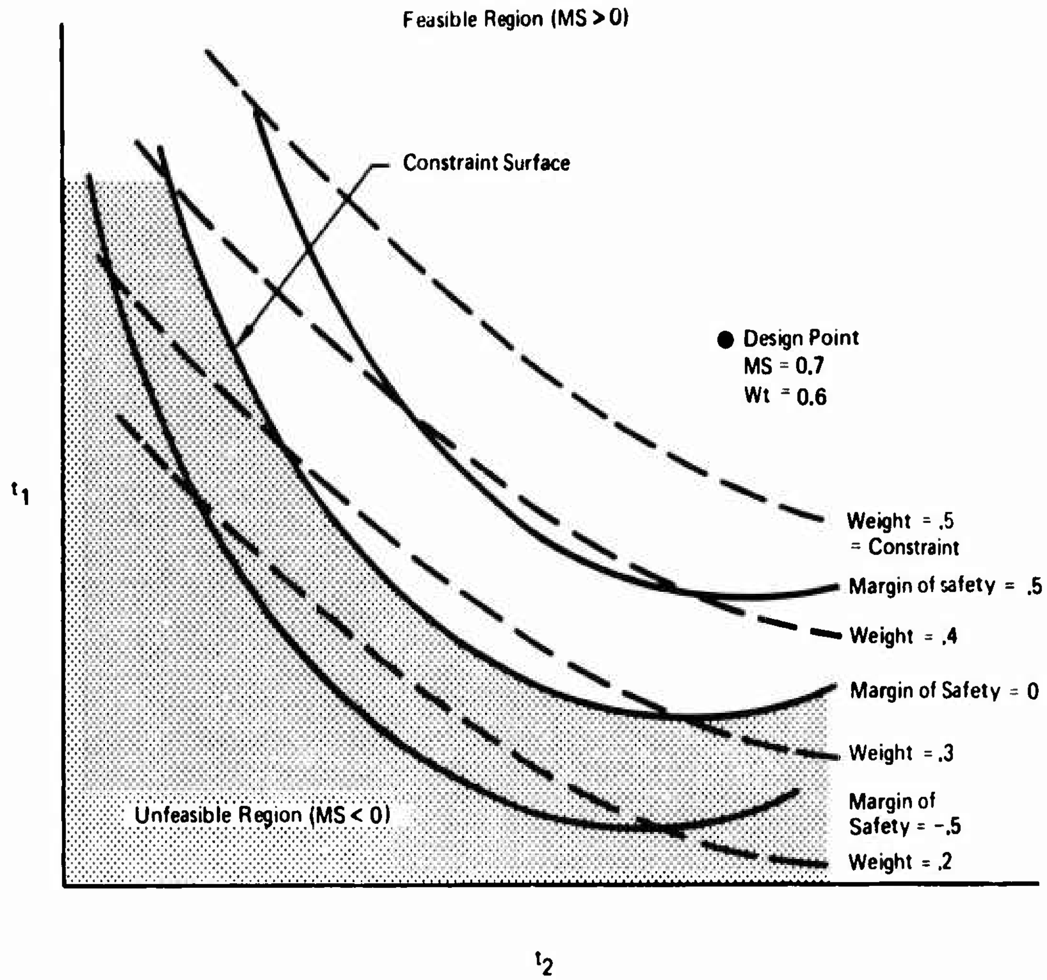


Figure 16. Design Space.

where MS_{BASE}, WT_{BASE}, and t_{iBASE} refer to the design point at the start of calculation of derivatives, before changing any thickness. Then these sets of partial derivatives can be used to calculate the required gradients.

$$\beta_{i_k} = \frac{\partial MS_k}{\partial t_i} \Big/ \sum_{j=1}^{n} \left(\frac{\partial MS_k}{\partial t_i}\right)^2 \qquad \begin{array}{l} i = 1,N \\ k = 1,M \\ M = \text{number of constraints} \end{array}$$

and

$$\alpha_i = \frac{\partial WT}{\partial t_i} \Big/ \sum_{j=1}^{n} \left(\frac{\partial WT}{\partial t_i}\right) \qquad i = 1,N$$

These gradients are actually the direction cosines of the normal to each curve at that design point.

For example, assume that the starting design point is chosen at Point A in Figure 17 and that the weight and margin of safety have been calculated for this point. As yet, no partial derivatives have been calculated, so the thickness will be decreased in steps in the same ratio to get to the constraint curve (MS=0). In the general case, the program changes only the thicknesses in first reaching the constraint curve, leaving the side, height, etc., dimensions constant. Then the program calculates the gradients of the weight and the margin-of-safety curves as previously described. There is a different margin-of-safety curve for each leg, since changing the thickness of one leg will change the margin of safety in that leg more than in another leg. This requires the calculation of a gradient of the margin of safety in each leg, as shown in Figure 18. A composite constraint gradient is computed using the gradient to each constraint curve and the margin of safety in each leg.

$$\{\gamma\} = \{\beta_1\} \ (1-MS_1)^{15} + \{\beta_2\} \ (1-MS_2)^{15} + \ldots + \{\beta_M\} \ (1-MS_M)^{15}$$

The proportion in which each constraint gradient is added into the composite constraint gradient is determined by the distance of the design point from that constraint.

In Figure 18 for case A, in which the design point is at the intersection of two constraints, each gradient is added in equally, resulting in a new direction. In case B, in which the design point is on one constraint and far from the other, the gradient of the constraint which the design is on will have the major influence on the new direction. This composite

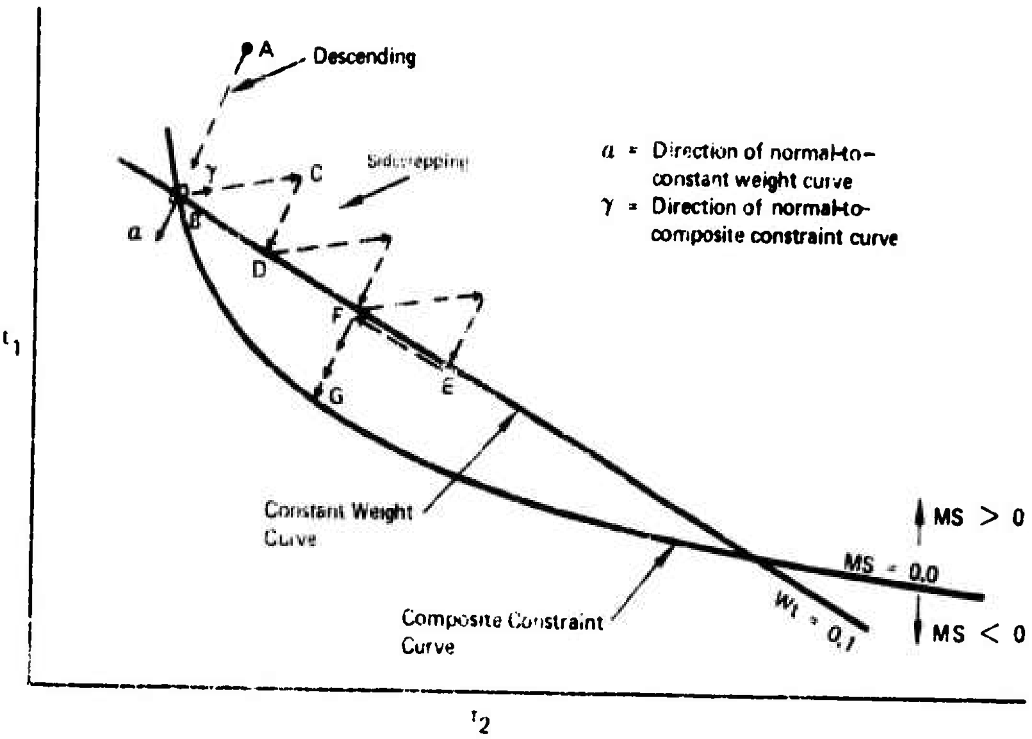


Figure 17. Two-Dimensional Optimization Example.

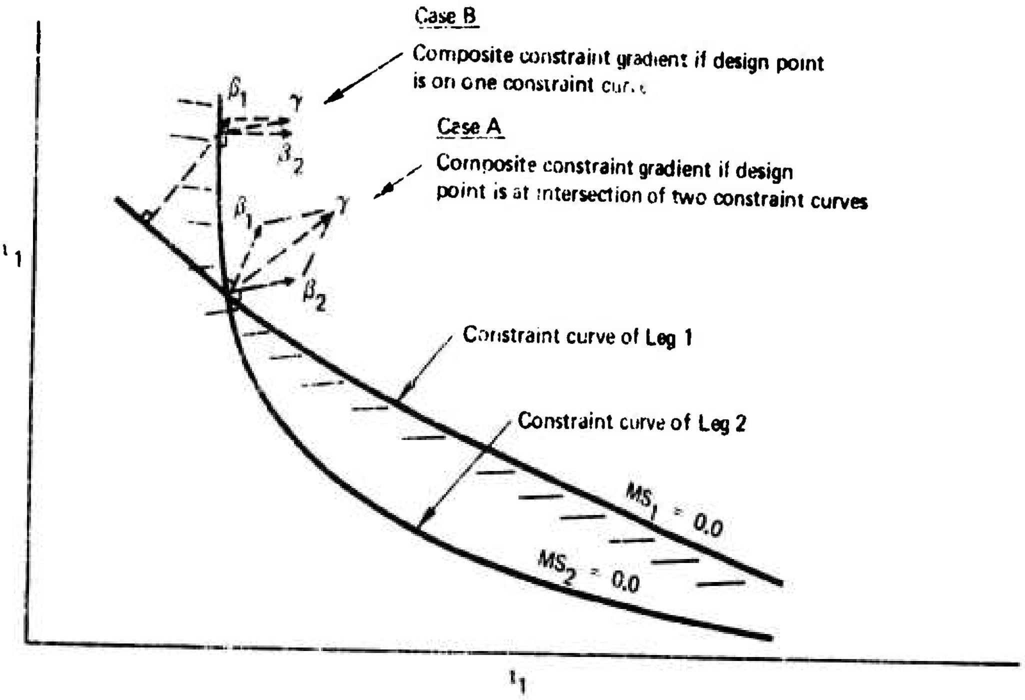


Figure 18. Composite Constraint Gradient.

constraint gradient is used by the program to make a step in the direction of maximum change in margin of safety, to Point C in Figure 17. The step is calculated:

$$t_{i+1} = t_i + \gamma_i X$$

where X is the step length. Next, the design variables are decreased along the gradient to the constant weight curve to reach the starting weight (Point D). This cycle B-C-D results in a sidestep away from the constraint curve in the direction of maximum increase in margin of safety, while maintaining a constant weight. The margin of safety is calculated at Point D. If the margin of safety is now greater than it was at Point B, then the bucket in the curve has not been reached and another sidestep is taken (D-F). Margin of safety is recalculated and another sidestep is taken (F-E). When the margin of safety is calculated at Point E, it is found to be less than at Point F. Therefore, the bucket in the constraint curve has been passed and a backward step is taken (E-F). Now the variables are decreased along the gradient to the constant weight curve (this is known as the direction of steepest descent) in steps, to reach the constraint curve (Point G). This series of operations (B-G) is called one cycle in the program. At the completion of one cycle (Point G), gradients are recalculated, the step length is reduced, and another series of sidesteps is begun. The number of optimization cycles to be run is input as "NSTP" on card type one. If a zero or blank is input for "NSTP," the program will set "NSTP" = 3. If the program makes ten sidesteps (B-D, D-F, etc.) without reaching the bucket, the program will descend to the constraint curve, recalculate gradients, and begin sidestepping again with a larger step length to reach the optimum in a fewer number of steps. At the end of each cycle (B-G), the weight of the design is checked. If the new weight is less than the weight at the end of the previous cycle, the variables are stored as a new optimum design. Generally, the program will find a series of designs, each lighter than the previous one. However, it is possible that the optimum will be found on the first or second cycle, and the rest of the cycles will result in heavier designs. This makes it necessary to store the lightest design found, since the program may not return to it.

In general, the constraint and constant weight curves are not two dimensional but are N-dimensional hypersurfaces, N being the number of variables. Figure 19 illustrates the set of constraint surfaces for a three-dimensional problem. Generally, several optimum designs will be possible for a given set of variables, and the program will only find the one that is

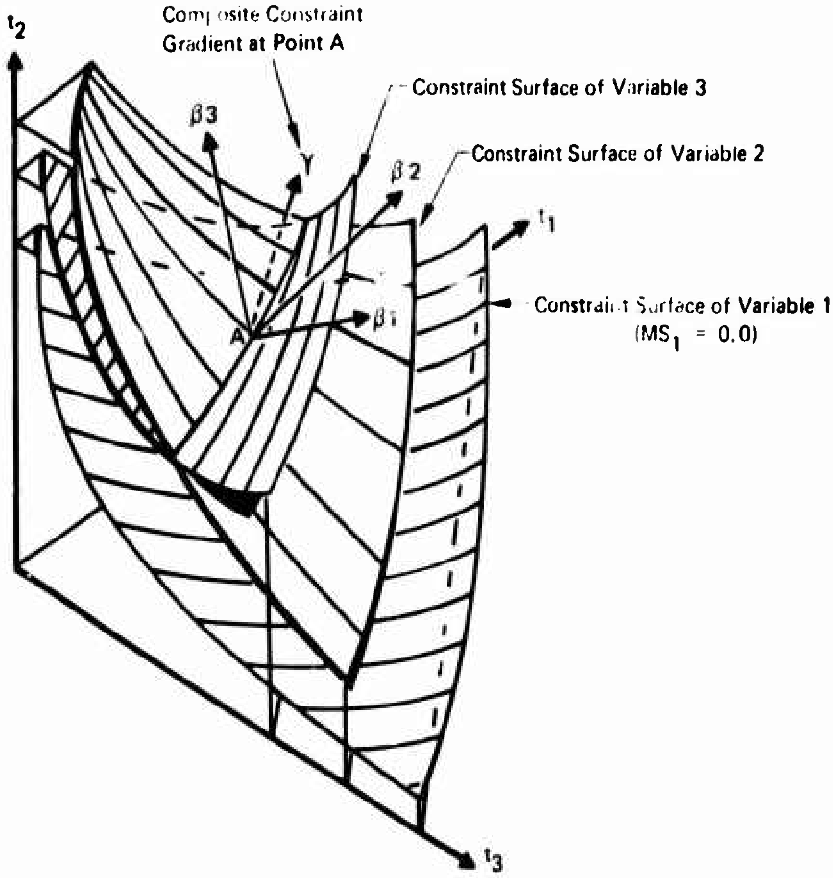


Figure 19. Constraint Surfaces of Three-Dimensional Problem.

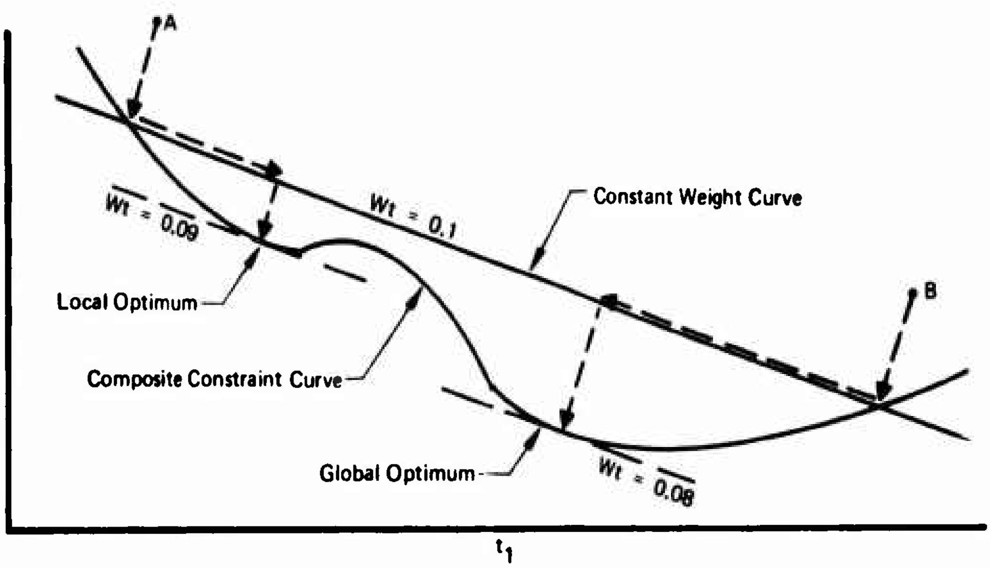


Figure 20. Global Optimum.

nearest to the input starting design. In Figure 20, if optimization is begun from Point A, an optimum design with a weight of 0.9 pound/square inch will be found, while starting at Point B will result in a weight of 0.08 pound/square inch. Thus it is necessary to make several optimization runs to determine if the "global" or most optimum design has been found.

The steepest descent/sidestep optimization method can be thought of as a way of getting down a mountain pass to the lowest altitude possible. The sides of the mountain represent constraints, and lines of constant altitude represent lines of constant weight. Once a given altitude has been reached by the design point, the altitude cannot be increased--only decreased.

A plot of the weight versus number of cycles for a typical case is shown in Figure 21. For this run a true Tetra-core flat plate was optimized with the side length held constant, the height and vertical leg thickness allowed to vary, and the Skew A and B leg thicknesses held the same. The largest change in weight was in the first cycle, indicating that the optimum region was found quickly, with the rest of the cycles being taken up with searching the immediate region to find the exact location of the optimum.

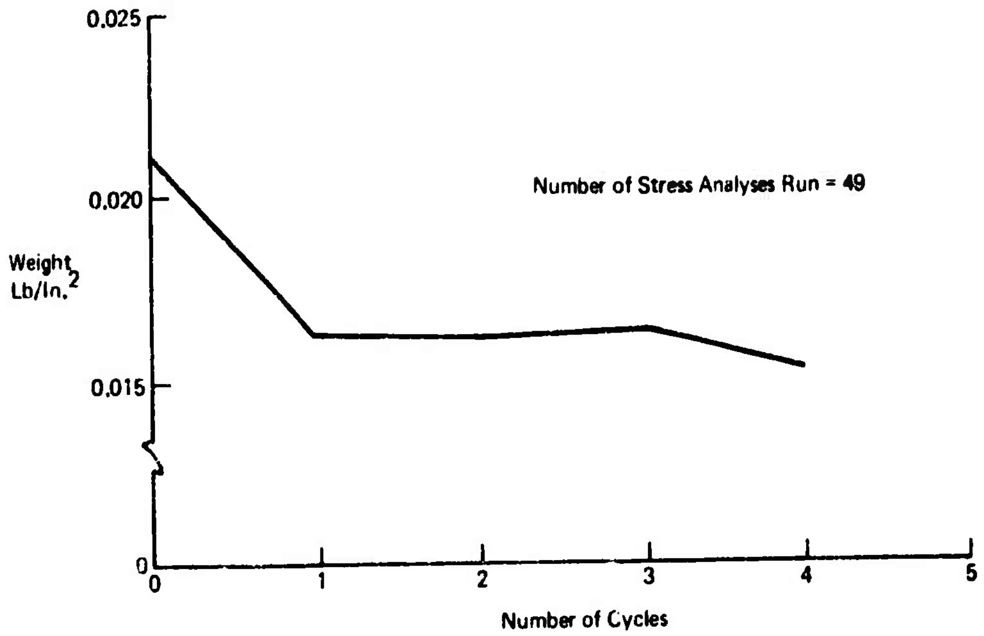


Figure 21. Example of Tetra-Core Optimization.

NONLINEAR ANALYSIS

An analysis of the effect of the nonlinear stress-strain behavior of the fiber-glass-epoxy composite used in Tetra-core legs can be conducted with this program. Stress-strain curves of the composite, determined by cutting out legs from the panel and testing them separately, are used as input, as is the loading increment to be applied[5]. First, the program calculates the stresses and strains in each plate, using the tangent moduli and Poisson's ratios input on Card Type 4 and the loads input on Card Types 12 and 13. Then, a new modulus is calculated for each plate, using the stress-strain curves input for each leg and the strains already calculated. As an example, suppose that the strain in the fiber direction (ε_1) of one plate had been + 0.0011 at the end of the first load increment, and that the tension stress-strain curve shown in Figure 22 had been input on Card Type 8 as a series of points for the leg which this plate was in, then the program takes the strain of 0.0011 and uses a linear curve fitting method to determine the stress corresponding to the given strain. This stress is used to compute a secant modulus to be used by the plate for the second iteration:

$$E_{sec} = \sigma_1/\varepsilon_1$$

The secant modulus is determined in the longitudinal, transverse, and shear directions from the stress-strain curves input for those directions. A different stress-strain curve is input for tension and compression for the longitudinal and transverse directions, and the correct one is chosen by referring to the sign of the strain in that direction.

Once moduli and Poisson's ratio in each plate have been determined, the loads are increased by the amount input on Card Types 12 and 13 and a new stress analysis is run. Then the stresses and strains in each plate are used to determine a new set of secant moduli and Poisson's ratio for each plate. The loads are increased again and the stresses are recalculated. This loading in increments continues until the number of steps specified on Card Type 1 as "NSTP" are completed. As the loads are increased, some plates will reach their failing strain in some direction. When this happens, the secant modulus in that direction is reduced to a small number, which unloads that plate without allowing the nodes to which it is attached to become free, as would happen if the modulus were reduced to zero. On the next loading increment, strains in the adjoining plates will be higher than would be caused by the increase in end load only, since the loads in the failed plate have been dumped into them. This will result in the failure of more plates in

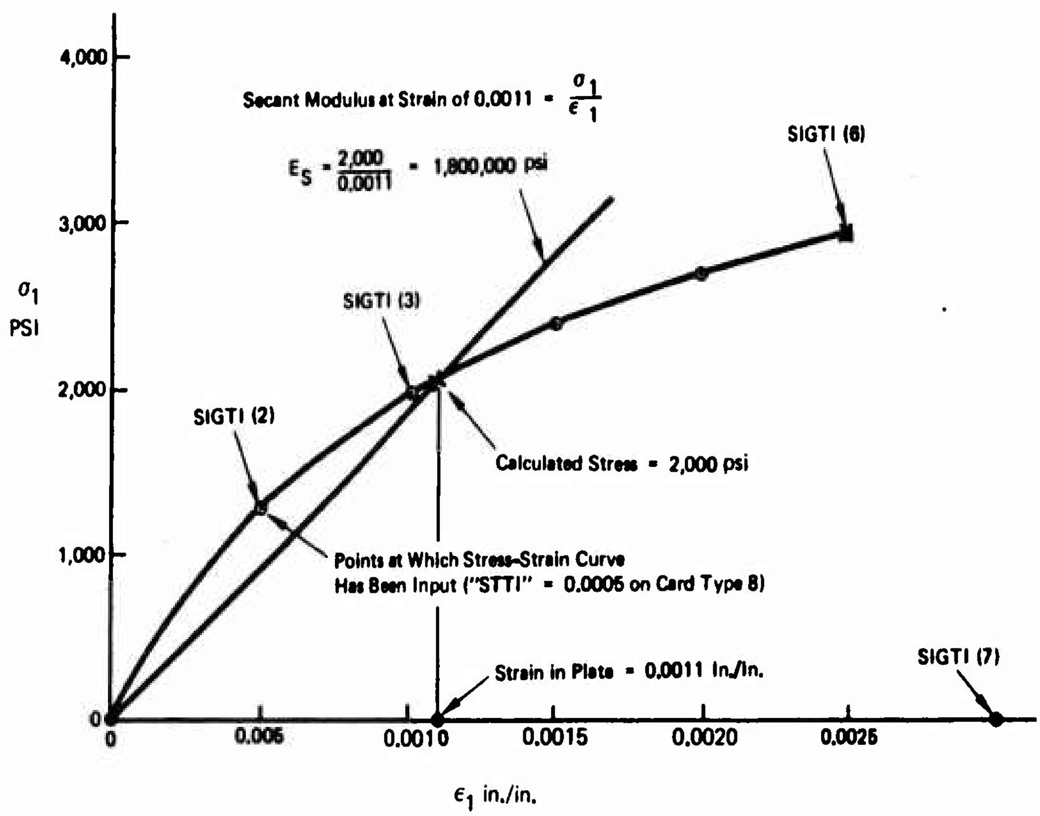


Figure 22. Method Used To Determine Moduli for Plastic Analysis.

the same area. Finally, enough plates will have failed that there is no load path through the structure, resulting in complete failure. Figure 23 illustrates a nonlinear analysis run on a Tetra-core panel with several plates removed to simulate the effect of a hole. Initially, all plates except those that have been removed are intact, but as the load is increased, several plates around the hole fail; and finally at a load of 3,000 pounds/inch, total failure occurs. If a lower increment of load had been used, the spread of failure around the hole would have been more visible.

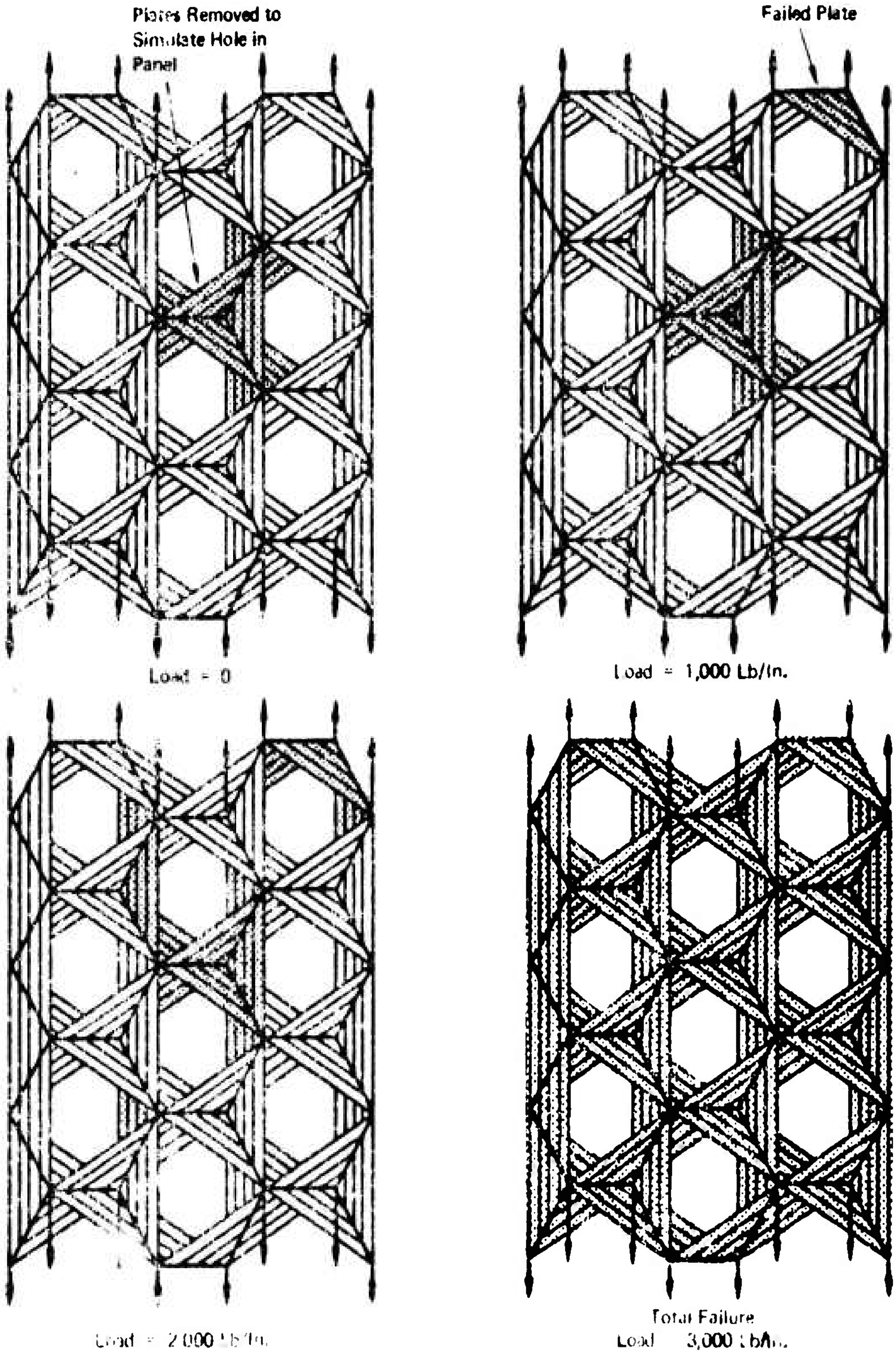


Figure 23. Example of Nonlinear Analysis.

BUCKLING ANALYSIS

A buckling analysis is performed on the plate elements in each leg making up the Tetra-core panels in the subroutine BUKL. No equation was available to calculate the compression buckling load of an orthotropic triangular plate with one edge free, loaded in compression. Therefore, it was decided to use the compression buckling equation (developed by Lackman and Ault[6]) for a rectangular orthotropic plate with one edge free and three edges simply supported. The length of the free edge on the triangle was used as the length of the rectangle, with the tetrahedron height used as a rectangle width. This simplification is expected to result in conservative values for the buckling load, due to the greater restraint imposed on the panel by the sides of the triangle compared with the sides of a rectangle. The equation used to calculate the buckling load is

$$\beta\left[\beta^2 + \left(\frac{\pi b}{a}\right)^2 \lambda\right]\left[\alpha^2 - \left(\frac{\pi b}{a}\right)^2 \mu_{xy}\right] \text{TANH}\alpha = \alpha\left[\alpha^2 - \left(\frac{\pi b}{a}\right)^2 \lambda\right]$$

$$\left[\beta^2 + \left(\frac{\pi b}{a}\right)^2 \mu_{xy}\right] \text{TAN}\beta,$$

where

$$\alpha = \left(\frac{\pi b}{a}\right)^{1/2}\left[\left(\frac{D_3}{D_2}\frac{\pi b}{a}\right)^2 - \frac{D_1}{D_2}\left(\frac{\pi b}{a}\right)^2 + \frac{N_x b^2}{D_2}\right]^{1/2} + \frac{D_3}{D_2}\left(\frac{\pi b}{a}\right)^{1/2}$$

$$\beta = \left(\frac{\pi b}{a}\right)^{1/2}\left[\left(\frac{D_3}{D_2}\frac{\pi b}{a}\right)^2 - \frac{D_1}{D_2}\left(\frac{\pi b}{a}\right)^2 + \frac{N_x b^2}{D_2}\right]^{1/2} - \frac{D_3}{D_2}\left(\frac{\pi b}{a}\right)^{1/2}$$

$$D_1 = \frac{E_x t^3}{12(1 - \mu_{xy}\mu_{yx})}$$

$$D_2 = \frac{E_y t^3}{12(1 - \mu_{xy}\mu_{yx})}$$

$$D_3 = \frac{2\, G_{XY} t^3}{12} + \frac{E_X \mu_{yx} t^3}{12(1 - \mu_{xy}\mu_{yx})}$$

$$\lambda = 4\, G_{xy}\,(1 - \mu_{xy}\mu_{yx})/E_y + \mu_{xy}$$

a = tetrahedron side length

b = tetrahedron height

t = leg thickness

An iteration method is used to find the lowest value of N_x which will satisfy the equality. This calculation is made for the Vertical, Skew A, and Skew B legs. If face sheets are applied to the Tetra-core panel, then all edges of the triangle will be simply supported and the equation of a simply supported rectangle is used[7].

$$N_x = 2\left(\frac{\pi}{b}\right)^2 \left[(D_1 D_2)^{1/2} + D_3 \right]$$

This equation is used for vertical, Skew A and B legs, and upper and lower face sheets when face sheets are applied. If a more accurate buckling equation becomes available, it can easily be inserted into the buckling subroutine.

MARGIN OF SAFETY

The margin of safety is calculated for each plate element using the von Mises-Hill-Tsai equation[8].

$$MS = 1/ \left(\frac{\sigma_1}{F_x}\right)^2 - \frac{F_y}{F_x}\frac{\sigma_1}{F_x}\frac{\sigma_2}{F_y} + \left(\frac{\sigma_2}{F_y}\right)^2 + \left(\frac{\tau_{12}}{F_{xy}}\right)^2 \quad -1$$

σ_1, σ_2, and τ_{12} are the stresses as in the local plate element coordinate system, with σ_1 always in the fiber direction. F_x, F_y, and F_{xy} are the allowable stresses, with F_x being the allowable stress in the fiber direction, F_y the transverse allowable, and F_{xy} the shear allowable. Tension or compression allowable stresses (as input on card type 10) are used, depending on the sign of the stress. A margin of safety is calculated for the four locations in the plate for which stresses have been calculated, with the lowest margin being printed out for that plate. The margin of safety in buckling is also calculated, using the buckling stress calculated for the leg in which this plate is found.

$$MS_B = \frac{F_c}{\sigma_1} - 1$$

The buckling margin is calculated only if one of the σ_1's is negative, indicating a compression load in the plate. The lowest margin of safety found in each leg for that load case is printed after printing of the plate element stresses.

CALCULATION OF ULTIMATE LOADS

After the margin of safety in each plate has been determined, the minimum margin of safety found in any plate is used to extrapolate the given applied load to determine the load at which the model would have failed.

$$X_{allow} = X_{load}/(1. - Y_{MS})^{.5}$$

$$Y_{allow} = Y_{load}/(1. - Y_{MS})^{.5}$$

$$XY_{allow} = XY_{load}/(1. - Y_{MS})^{.5}$$

$$XM_{allow} = X_{mom}/(1. - Y_{MS})^{.5}$$

$$YM_{allow} = Y_{mom}/(1. - Y_{MS})^{.5}$$

$$XYM_{allow} = XY_{mom}/(1. - Y_{MS})^{.5}$$

$$TORQ_{allow} = TORQ/(1. - Y_{MS})^{.5}$$

$$XQ_{allow} = XQ_{SHR}/(1. - Y_{MS})^{.5}$$

$$YQ_{allow} = YQ_{SHR}/(1. - Y_{MS})^{.5}$$

where Y_{MS} is the minimum margin of safety found in any plate. X_{load}, Y_{load}, etc., are the input loads from card types 12 and 13. The power of 0.5 may have to be changed to agree with tests when they become available.

CALCULATION OF STIFFNESSES

An equivalent modulus in the x direction is calculated whenever a uniform load is applied in that direction. This is done by taking two x deflections on opposite ends of the panel and adding them to determine the total change in length of the panel (ΔX). This ΔX is divided by the original panel length to get an equivalent strain $\varepsilon_{x_{EQ}}$. The uniform load on the x face of the panel, divided by the height of the face, gives an equivalent stress $\sigma_{x_{EQ}}$. The equivalent modulus in the x direction is then

$$E_{x_{EQ}} = \frac{\sigma_{x_{EQ}}}{\varepsilon_{x_{EQ}}}$$

An equivalent modulus in the y direction is calculated in the same way whenever a uniform load is applied to the y face.

OUTPUT DESCRIPTION

Primary output of the Tetra-core analysis program consists of the finite-element model generated by the data generator overlay and the deflections and stresses calculated by the stress analysis overlay. If the optimization option is used, additional printout is generated during each step of optimization.

The printed output from a standard stress analysis run starts with the input data (number of load cases, size of tetrahedron, leg thicknesses, etc.); the printout of calculated data starts with the number of node points and plate elements generated by the data generator. This node point number does not include the nodes at the midpoint of the triangle or quadrilateral which are added in the stress analysis overlay. The number of load cases is also printed. If a cylinder is generated, the radius to the outer surface is printed. If an airfoil is generated, the chord length is printed. The X, Y, Z coordinates of the generated nodal points are printed. Nodal loads are printed for each load case along with the original node number and the new node numbers that include mid-side nodes added by each linear strain triangle and quadrilateral. Each linear strain element used in the model adds a new node point at the midpoint of two sides of the element where it is attached to another element, which must be accounted for by increasing the size of the load vector and the stiffness matrix. The new node numbers are generated from the node numbers of the vertices on each side of the new node (Figure 24). The new node number is generated by multiplying the larger of the two nodes by 1,000 and adding to the smaller, resulting in a unique new node number. For example, if a new node is to be generated between nodes 5 and 3, as shown in the figure, the resulting number would be 5003. This node is then added into the stiffness matrix and load vector behind the location of the larger of the two vertex nodes. The added mid-side nodes for a typical case are shown in Figure 26.

Plate properties are printed. The node numbering sequence for each plate in the original node point order and as renumbered is printed. Nodes listed under the headings I, J, K, and L give the original plate connectivity. Nodes listed under LP, LQ, LR, LS, LT, and LU give the connectivity using the new nodal numbering sequence. LS and LT represent new mid-side nodes for the triangle. Plate moduli and plate thickness are printed along with the node numberings. EX, EY, and EXY have been divided by the factor $1-\mu_{xy}\mu_{yx}$ for use in the element stiffness generator. EX is the Young's modulus in the direction of nodes I and J, and EY is at an angle of 90 degrees to EX.

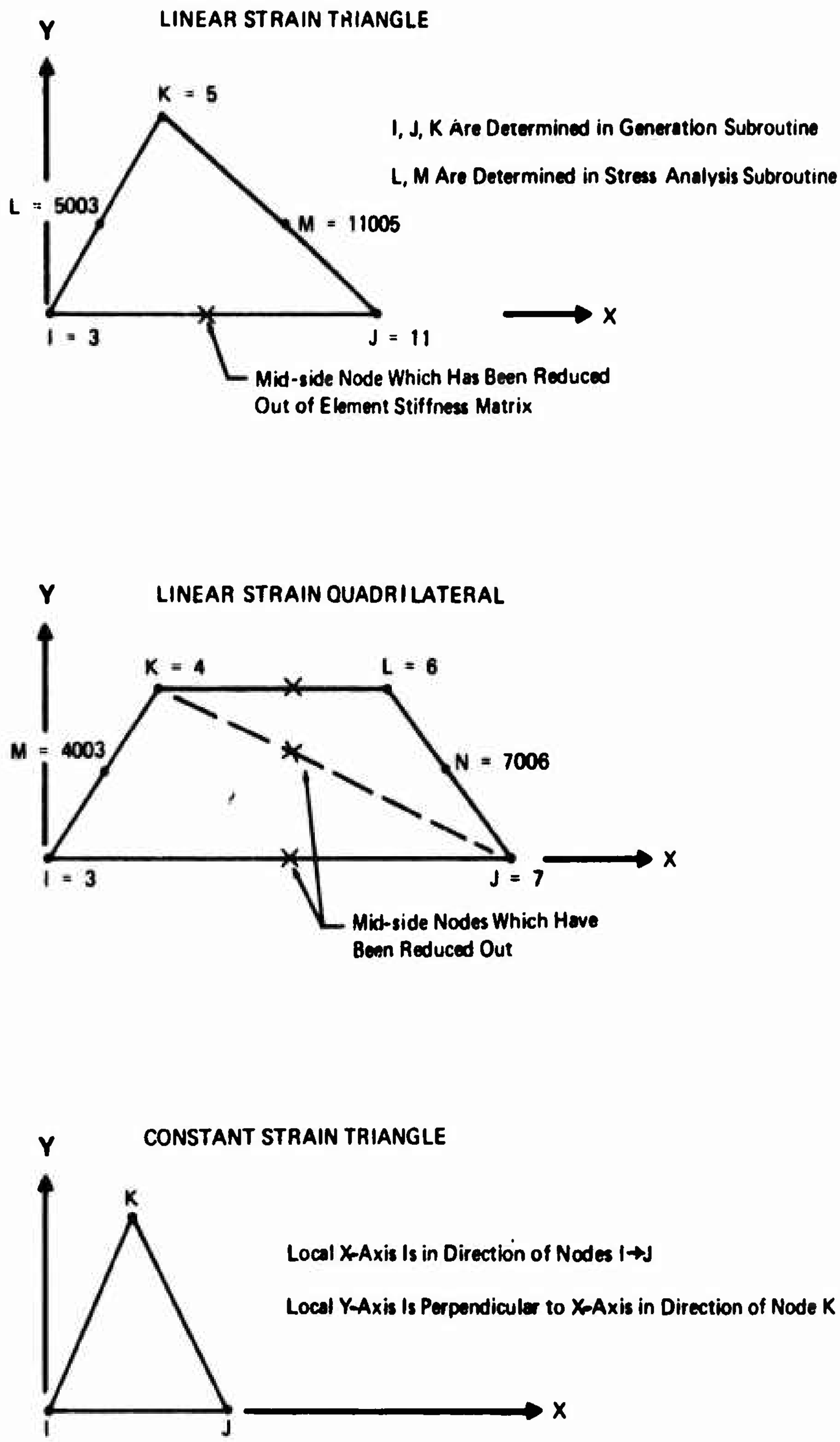


Figure 24. Elements Used in Tetra-Core Program.

Number of Vertex Nodes	=	72
Number of Mid-Side Nodes	=	96
Total Number of Nodes	=	168

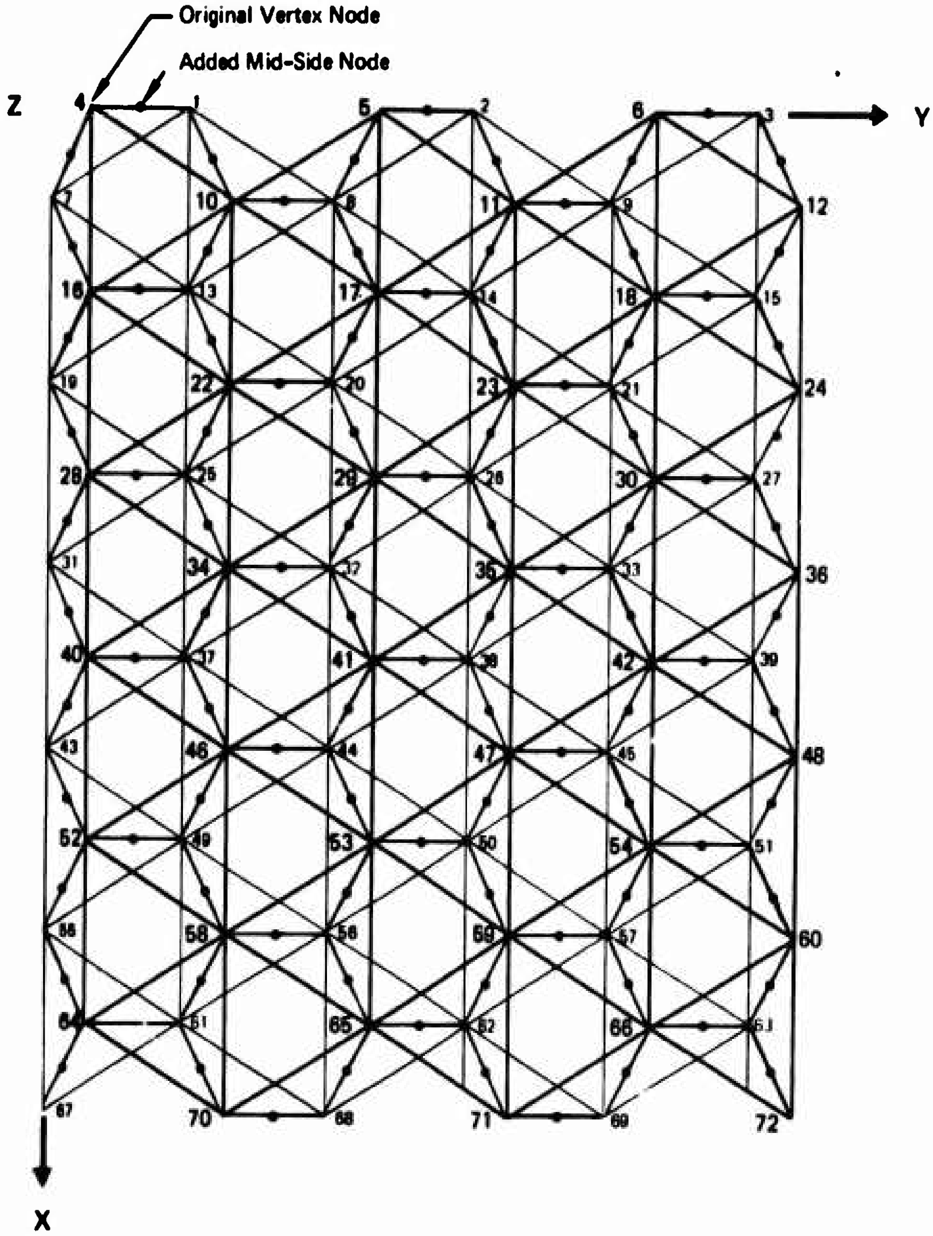


Figure 25. Added Mid-Side Nodes, True Tetra-Core Flat Plate $L_X = 6$, $L_Y = 6$.

The 1/2 bandwidth, in nodes, required to store the stiffness matrix is printed after the elements have been renumbered and printed. The storage space required for the overall stiffness matrix can be calculated by multiplying the number of nodes times 3 times the 1/2 bandwidth times 3. If this number added to the amount of space required to store the loads vectors is greater than the available disc space, then the problem will have to be reduced in size.

Nodal deflections and plate stresses are printed for each load case separately. Deflections are printed for each node with the node points numbered in the original sequence (1, 2, 3, 3001, 4, 4002, etc.). Stresses are printed for four locations in the linear strain element used in Vertical and Skew A and Skew B legs, and for the triangle center for the constant strain triangle used in face sheets. The connectivity of each triangle is printed in the original numbering sequence. For a linear strain triangle, the stresses are printed at node I, node J, node K, and at the center of the triangle, in that order. For a linear strain quadrilateral, which is used only in the truncated Tetra-core model, stresses are printed at nodes I, J, K and L. The margin of safety, calculated using the Hill-Tsai failure criterion, in each element and the leg which that element is in (Vertical leg = leg 1, Skew A = leg 2, Skew B = leg 3, lower face = leg 4, upper face = leg 5) are printed also. The buckling stress of a triangle in each leg is printed.

After all the plate stresses in a load case have been printed, the minimum margin of safety found in each leg is printed. Then the minimum margin of safety is used to extrapolate the original loads to predict a load case with the same ratio between each load that would have a margin of safety of zero. If the model has been loaded in the X or Y direction, the deflections of two node points on opposite ends of the model are used to calculate an equivalent modulus. Then printing of the deflections and stresses for the next load case starts.

If the nonlinear option is specified, a series of load increments will be applied until overall failure occurs. Nodal coordinates and load vectors will be printed out only for the first increment, since the coordinates do not change and the succeeding load vectors will be integer multiples of the first load vector. The load intensity (i.e., load in X direction in pounds/inch) is printed at the beginning of each load increment. Plate element moduli are printed at the start of each load increment to allow an inspection of the moduli in each plate to determine the extent of degradation in the panel. As each plate fails in a given direction (longitudinal, transverse, or shear), the modulus in that direction is reduced to near zero. Final failure occurs when all plates have failed. Deflections and stresses are printed out for each load increment.

If the hole option is specified, the moduli of each triangle that connects to a node in the center of the plate, cylinder, or airfoil are set to zero. This node is normally determined by the program but can be input if desired. The identity of the affected node is printed out in either case.

If the optimization option is specified, the coordinates, loads, elements, deflections, and stresses are printed out for the input configuration and for the final optimized configuration only. Intermediate output during each optimization step consists of the current design configuration and the margin of safety in each leg. When partial derivatives are being calculated, the variable that is being changed is printed out as "IK". Direction cosines of the constant weight surface, direction cosines of the constraint surface of each leg, margin of safety in each leg, and direction cosines of the composite constraint surface are printed at the start of each cycle of sidestepping. Weight of the design is checked at the end of each sidestepping cycle to determine if a new optimum has been reached. If so, the design variables are printed and optimization continues. When the input number of sidestep cycles "NSTP" have been completed, the optimization process is stopped and the deflections and stresses for the lowest weight design that was found are printed.

SAMPLE PROBLEMS

Examples of input for several types of analyses will be given for reference. For a nonlinear analysis, the set of cards shown in Figure 26 will be required. Three sets of cards of Types 8 and 9 are required to input stress-strain curves for each leg. The input cards required for an airfoil analysis are shown in Figure 27. An example of the input for an optimization run is shown in Figure 28.

The output from an optimization run is shown in Appendix II.

```
   1    0    1    1    0    1                                                       - - - - Card Type 1
  TETRA-CORE PLATE STRESSES                                                         - - - - Card Type 3
4000000.     1700000.     930000.      .25                                            - -
4000000.     1700000.     930000.      .25                                            - -  } Card Type 4
4000000.     1700000.     930000.      .25                                            - -
   4    4    1.0    0.0   .1     .1    .1                                           - - - - Card Type 5, 6
.0025        .002         .0025        .002         .005
0.0          8500.        16000.       23000.       31000.       37500.       0.0
0.0          3300.        6600.        9700.         12700.      0.0          0.0
0.0          12000.        23400.        33000.     41500.       47500.       0.0
0.0          3300.        6600.        9700.         12700.      0.0          0.0
0.0          4400.        7000.         9000.       10600.      0.0           0.0
.15          .142         .136         .128         .120         .113         0.0
.280         .265         .252          .240        .227         .214         0.0
.0025        .002         .0025        .002         .005
0.0          8500.        16000.       23000.       31000.       37500.       0.0
0.0          3300.        6600.        9700.         12700.      0.0          0.0
0.0          12000.        23400.        33000.     41500.       47500.       0.0    } Card Type 8, 9
0.0          3300.        6600.        9700.         12700.      0.0          0.0
0.0          4400.        7000.         9000.       10600.      0.0           0.0
.15          .142         .136         .128         .120         .113         0.0
.280         .265         .252          .240        .227         .214         0.0
.0025        .002         .0025        .002         .005
0.0          8500.        16000.       23000.       31000.       37500.       0.0
0.0          3300.        6600.        9700.         12700.      0.0          0.0
0.0          12000.        23400.        33000.     41500.       47500.       0.0
0.0          3300.        6600.        9700.         12700.      0.0          0.0
0.0          4400.        7000.         9000.       10600.      0.0           0.0
.15          .142         .136         .128         .120         .113         0.0
.280         .265         .252          .240        .227         .214         0.0
38000.       13600.       47400.       13600.       11000.                              - - -
38000.       13600.       47400.       13600.       11000.                              - -  } Card Type 10
38000.       13600.       47400.       13600.       11000.                              - -

1000.                                                                               - - - - Card Type 12
                                                                                        - - Card Type 13 (Blank)
```

Figure 26. Input Data for Nonlinear Analysis.

```
   3    0    0    3    0    0    0    0    0
   TETRA-CORE AIRFOIL
4000000.      1700000.      930000.       .25
4000000.      1700000.      930000.       .25
4000000.      1700000.      930000.       .25
   4    4     1.0    .5     .1     .1     .1
0.0           0.0
0.05          0.10
.010          0.20
0.15          0.30
0.20          0.40
0.25          0.50
0.20          0.60
0.15          0.70
0.10          0.80
0.05          0.90
0.0           1.00
180000.       13600.        47400.        13600.        11000.
180000.       13600.        47400.        13600.        11000.
180000.       13600.        47400.        13600.        11000.
1000.
                                          1000.
                                                                      1000.
```

Card Type 1

Card Type 3

Card Type 4

Card Type 5, 6

Card Type 7

Card Type 10

Card Type 12, 13

Figure 27. Input Data for Airfoil Analysis.

```
     1     1     0     1     0     0   ---Card Type 1
    12
     5
     4                                   >Card Type 2
    12
    12
    12
    12
    12
    12
   TETRA-CORE PLATE STRESSES  -Card Type 3
  4000000.      500000.       500000.       .25
  4000000.      500000.       500000.       .25  >Card Type 4
  4000000.      500000.       500000.       .25
     4     4    .75    .50    .030   .030   .030 - Card Type 5
                                                 - Card Type 6 (Blank)
  11000.        13600.        11000.        13600.      11000.
  11000.        13600.        11000.        13600.      11000.  >Card Type 10
  11000.        13600.        11000.        13600.      11000.
  .1            .1            .1                          - - - Card Type 11
  1900.                       130.                        - - - Card Type 12
                                                          - - - Card Type 13 (Blank)
```

Figure 28. Input Data for Optimization Run.

COMPARISON WITH UNIVERSITY OF KANSAS TESTS

Computer runs have been made to compare to several of the tests run at the University of Kansas[9]. The results of this comparison are not very meaningful, however, because the basic properties of the material making up the legs were not specified in the test document. Information in the test document was limited to side length and tetrahedron height and the fiber used, which was S glass. The program requires elastic moduli and thickness for each leg. It was assumed that the material had the following moduli and thickness:

$$E_X = 4{,}000{,}000 \text{ psi}$$

$$E_Y = 500{,}000 \text{ psi}$$

$$G_{XY} = 500{,}000 \text{ psi}$$

$$\mu_{XY} = 0.25$$

$$t = 0.030 \text{ in.}$$

Using these as input with a side length of 0.6123 inch and a height of 0.4 inch (comparable to specimen 1 in Reference 9) produced an E_X of 281,000 psi and an E_Y of 284,000 psi. The predicted initial failure load, using a linear stress analysis, was 789 pounds. The cause of this failure was a high transverse tension stress in some Skew A and B plates which would not have resulted in final failure. If the criterion of failure had been fiber stress, the predicted ultimate load would have been 3,300 pounds for a fiber stress of 600,000 psi. This compares to an E_X of 363,053 psi, an E_Y of 303,198 psi, and an ultimate load of 5,800 pounds from the test. However, a comparison between the analysis and test would be risky, since material properties of the specimens used for the test were guessed at. The low predicted load of 789 pounds was caused by failure of the Skew A and B legs in the transverse direction.

An analysis run with a configuration comparable to that of Specimen 3 (HT = 0.5, side = 0.75) was run, resulting in an E_X of 293,400 psi that compares to the test result of E_Y = 288,150 psi. This appears to be good agreement, assuming that the input moduli and thickness were close.

MODEL CONSTRAINTS

The Tetra-core analysis program is dimensioned for 1100 plate elements, 300 vertex nodes, and 600 total nodes. A problem of this size cannot be run on the IBM 360/44 due to a lack of disc space. However, the program can be converted to a larger 360, such as a 360/65, very easily, should one become available. This can be done by removing the call load cards which are used to load a new overlay before referring to it. These cards are not necessary on the operating system used by the larger IBM machines. Also, job control cards must be changed. With a larger machine it will be possible to increase the size of problem that may be run to 600, or greater vertex nodes. This can be done by increasing the X, Y, Z and UX, UY, UZ dimensions in the generation and stress analysis subroutines. The dimensions of variables A and DIAG in subroutine STIFF must be increased. In subroutine SOLPAC, variables DIAG, AI, AJ, B, and X must be increased in size. In subroutine DEFL, variables IDO and B must be increased in size.

In converting to a CDC 6600 machine, it will be necessary to put in OVERLAY and CALL OVERLAY cards for each overlay. Direct access read/write cards used for the IBM 360 will not work on the CDC 6600. Direct-access read/write commands will have to be rewritten for the particular CDC 6600 on which the program is to be run, since they vary from installation to installation.

The truncated Tetra-core model can generate the full range of Tetra-core geometries, from a true Tetra-core to a truncated Tetra-core with vertical sides. However, both the true Tetra-core and the vertical-sided Tetra-core models will have several nodes which are connected to each other by plates generated for the same X, Y, Z coordinates. This will cause a linear dependence in the stiffness matrix, since these nodes will essentially have the same stiffnesses; that is, their rows and columns will be identical. This will result in the matrix being singular, and no solution can be found. This problem will also result when the connected nodes are within a small distance of each other.

The program does not compute nodal reaction force, since this would require more disc storage space than is available on the IBM 360/44 at Fort Eustis. If the program is converted to a larger machine, reactions can be calculated by storing a copy of the merged stiffness matrix and then multiplying it by the calculated deflections: $P = K\Delta$.

COMPUTER PROGRAM

Input Formats for Tetra-Core Analysis Program

CARD TYPE	VARIABLE	FIELD	COMMENT
1	NLOD	I (1-4)	Number of load cases to be run. If "IOPT" = 1 or "NLIN" = 1 then NLOD must = 1.
	IOPT	I (5-8)	If "IOPT" = 0, no optimization will be done. If "IOPT" = 1, the Tetra-core element will be optimized for the given load case.
	NLIN	I (9-12)	If "NLIN" = 0, standard linear stress analysis will be run. If "NLIN" = 1, a nonlinear analysis will be run using the input load case as the load increment and "NSTP" as the number of iterations.
	ITYP	I (13-16)	If "ITYP" = 1, flat plate analysis If "ITYP" = 2, cylinder analysis If "ITYP" = 3, airfoil analysis
	IFACE	I (17-20)	If "IFACE" = 0, no face sheets will be added to the Tetra-core. If "IFACE" = 1, face sheets will be added.
	IHOLE	I (21-24)	If "IHOLE" = 0, a standard analysis will be run. If "IHOLE" = 1, all plates connected to a grid point in the center of the panel will be defined with zero stiffnesses, giving the effect of a hole in the panel. If "IHOLE" > 1, all plates connected to grid point "IHOLE" will have zero stiffness, allowing a hole to be inserted where desired.
	NSTP	I (25-28)	"NSTP" determines the number of iterations to be made in a nonlinear analysis, or the number of optimization cycles to be made in an optimization run. The default option for "NSTP" is 3.

CARD TYPE	VARIABLE	FIELD	COMMENT
	ITET	I (29-32)	If "ITET" = 0, standard tetrahedrons will be generated. If "ITET" = 1, truncated tetrahedrons will be generated.
	ITEST	I (33-36)	If "ITEST" = 1, the element and merged stiffness matrices will be printed out. If "ITEST" = 0, no printout of matrices.
	ILOD	I (37-40)	If "ILOD" = 0, the program will automatically generate nodal loads and fixities. If "ILOD" > 0, nodal loads and/or fixities are input on Card Types 14 and 15, as shown below.

ILOD	Comment
1	Nodal loads are input on Card Type 14. Nodal fixities are automatically generated.
2	Nodal loads are input on Card Type 14. Nodal fixities are input on Card Type 15.
3	Nodal loads are automatically generated. Nodal fixities are input on Card Type 15.

CARD TYPE	VARIABLE	FIELD	COMMENT
2	ISA	I (1-4)	Card Type 2 is input only if "OPT" = 1. "ISA", "ISB", "ISC" are used to set variables equal to each other during optimization. 11 cards of type 2 are input.
	ISB	I (1-5)	
	ISC	I (9-12)	During optimization the program will normally change each variable independently. ISA, ISB, ISC can be used to keep several variables, such as leg thicknesses, the same during optimization. ISA can also be used to hold variables constant during optimization by inputting ISA = 12 for that variable. One Card Type 2 is input for each variable, resulting in eleven cards of type 2. The first Card Type 2 is used to control the side length, the second controls height, the third controls vertical leg thickness, etc., as shown below. If ISA = 2 on the first card type 2 and ISA = 1 on the second card type 2 then the first and second variables (side length and height) will be kept the same during optimization. The program changes only the leg thicknesses during the first descent cycle when going from the input design to the initial design on the constraint curve. Therefore, at least one leg thickness must be allowed to vary during optimization or the program will not be able to find an initial feasible design point since it will not have anything to vary.

CARD TYPE 2 NUMBER	VARIABLE CONTROLLED	ISA	ISB	ISC
1	Side Length	12		
2	Height	12		
3	Vertical Leg Thickness	4	5	
4	Skew A Leg Thickness	3	5	

CARD TYPE 2 NUMBER	VARIABLE CONTROLLED	ISA	ISB	ISC
5	Skew B Leg Thickness	3	4	
6	Upper Face Sheet Thickness	12		
7	Lower Face Sheet Thickness	12		
8	THETA	12		
9	Y1OFF	12		
10	Y2OFF	12		
11	THT	12		

CARD TYPE	VARIABLE	FIELD	COMMENT
3	Title	A(1-48)	Title Card

CARD TYPE	VARIABLE	FIELD	COMMENT
4	EX	F(1-12)	Elastic modulus in longitudinal direction of leg, psi.
	EY	F(13-24)	Elastic modulus in transverse direction of leg, psi.
	GXY	F(25-36)	Shear modulus of leg,psi.
	μXY	F(37-48)	Poisson's Ratio
			One card type 4 must be input for each leg (Vertical Skew A, Skew B) and for the upper and lower faces if "IFACE" = 1.

CARD TYPE	VARIABLE	FIELD	COMMENT
5	LX	I(1-4)	Number of legs in X face of Tetra-core "LY" times the length of a tetrahedron times 0.866 gives the panel width.
	LY	I(5-8)	Number of legs in Y face of Tetra-core "LY" times the length of a tetrahedron side gives the panel length.
	SIDE	F(13-18)	Length of a tetrahedron side, in.
	HT	F(19-24)	Height of Tetra-Core Panel, in.
	THK(1)	F(25-30)	Thickness of Vertical Legs, in.
	THK(2)	F(31-36)	Thickness of Skew A Legs, in.
	THK(3)	F(37-42)	Thickness of Skew B Legs, in.
	THK(4)	F(43-48)	Thickness of Lower Face Sheet, in.
	THK(5)	F(49-54)	Thickness of Upper Face Sheet, in.
			"THK(4)" and "THK(5)" are input only if "IFACE" = 1 on card type 1.

CARD TYPE	VARIABLE	FIELD	COMMENT
6	THETA	F(1-12)	The variables input on this card control the variation of of the Tetra-core element from a true tetrahedron. See Figure 4 for a description.
	Y1OFF	F(13-24)	
	Y2OFF	F(25-36)	
	THT	F(37-48)	

CARD TYPE	VARIABLE	FIELD	COMMENT
7	TC	F(1-12)	Thickness/Chord Ratio
	XC	F(13-24)	X/Chord Ratio
			Card Type 6 is input only if "ITYP" = 3 on card 1, that is, only if an airfoil shape is to be generated. Eleven cards of type 6 must be input, each with a thickness/chord for the specified X/Chord. The X/Chord at the nose of the airfoil is input first, with X/Chord for the tail of the airfoil last. The intermediate X/Chord cards must be in sequence from nose to tail.

CARD TYPE	VARIABLE	FIELD	COMMENT
8	STT1	F(1-12)	Longitudinal tension strain increment, in./in.
	STT2	F(13-24)	Transverse tension strain increment, in./in.
	STC1	F(25-36)	Longitudinal compression strain increment, in./in.
	STC2	F(37-48)	Transverse compression strain increment, in./in.
	STSS	F(49-60)	Shear strain increment, in./in.
			"STT1", "STT2", "STC1", "STC2", "STSS" are the strain increments used in inputting tension and compression stress-strain curves on cards of Type 9. The stress-strain curves are broken down into increments of "STT1," "STT2," etc., for inputting on card 9.

CARD TYPE	VARIABLE	FIELD	COMMENT
9	SIGT(I) I = 1,7	F(1-12) F(13-25)... F(61-72) F(73-80)	Stress-strain curves of the material used in each leg and each face sheet are input if a nonlinear analysis is performed ("NLIN" = 1, card 1). Longitudinal tension stresses are input for strains of 0, STT1, 2STT1, 3STT1,...6STT1, for example. The first card following card 8 must contain the longitudinal tension stresses; the second card contains the transverse tension stresses, etc. The complete sequence is shown below.

CARD TYPE	VARIABLE	CONTENTS
8		Strain increments - "STT1," "STT2," "STC1," "STC2," "STSS.", in./in.
9	SIGT1	Longitudinal tension stresses in increments of strain "STT1.", psi.
9	SIGT2	Transverse tension stresses in incr nts of strain "STT2.", psi.
9	SIGC1	Longitudinal compression stresses in increments of strain "STC1.", psi.
9	SIGC2	Transverse compression stresses in increments of strain "STC2.", psi.
9	PRXT	Poisson's ratio μXY in tension in increments of "STT1."
9	PRXC	Poisson's ratio μXY in compression increments of "STC1."

One set of Card 8 + six card 9's must be input for each leg of the Tetra-core. If "IFACE" = 0 on card 1, then three sets must be input - Vertical legs, Skew A legs, and Skew B legs. If "IFACE" = 1, then five sets must be input- three sets for the legs, plus one set for the lower face sheet, and one set for the upper face sheet.

CARD TYPE	VARIABLE	FIELD	COMMENT
10	FXT	F(1-12)	Longitudinal tension allowable stress, psi.
	FYT	F(13-24)	Transverse tension allowable stress, psi.
	FXC	F(25-36)	Longitudinal compression allowable stresses, psi.
	FYC	F(37-48)	Transverse compression allowable stress, psi.
	SSS	F(49-60)	Shear Allowable Stress, psi.
			One card type 9 must be input for each leg. If "IFACE" = 1, five card 9's are input, one each for the Vertical legs, Skew A legs, Skew B legs, lower face sheet, upper face sheet. If "IFACE" = 0 only the first three are input.

CARD TYPE	VARIABLE	FIELD	COMMENT
11	RHO(1)	F(1-12)	Card 10 is input only if "IOPT" = 1 on card 1. Density of material used in vertical legs, $lb/in.^3$.
	RHO(2)	F(13-24)	Density of material used in Skew A legs, $lb/in.^3$.
	RHO(3)	F(25-36)	Density of material used in Skew B legs, $lb/in.^3$
	RHO(4)	F(37-48)	Density of material used in lower face sheet, $lb/in.^3$.
	RHO(5)	F(49-60)	Density of material used in upper face sheet, $lb/in.^3$.
			The density is used to calculate the weight of the Tetra-core for use in optimization.

CARD TYPE	VARIABLE	FIELD	COMMENT
12	XLOAD	F(1-12)	Load on XFACE of flat plate Tetra-core, lb/in.; or total end load on cylinder or airfoil, lb.
	YLOAD	F(13-24)	Load on YFACE of flat plate Tetra-core, lb/in.
	XYLOD	F(25-36)	Shear load on flat plate Tetra-core, lb/in.
	XMOM	F(37-48)	Moment on XFACE of flat plate Tetra-core in.-lb/in.; or total moment on cylinder or airfoil, in.-lb.
	YMOM	F(49-60)	Moment on YFACE of flat plate Tetra-core in.-lb/in.
	XYMOM	F(61-72)	Twisting moment on flat or plate Tetra-core in.-lb/in.
	TORQ	F(73-80)	Torque on cylinder or airfoil of Tetra-core, in.-lb/in.

CARD TYPE	VARIABLE	FIELD	COMMENT
13	XQSHR	F(1-12)	Shear load on XFACE of flat plate Tetra-core,lb/in.
	YQSHR	F(13-24)	Shear load on YFACE of flat plate Tetra-core,lb/in.
			There is an upper limit of 10 on the number of load cases that may be run.
			If a nonlinear analysis is to be made (NLIN" = 1) or an optimization is to be run ("IOPT" = 1), only one load case is allowed.

CARD TYPE	VARIABLE	FIELD	COMMENT
14	J	I (1 - 4)	Card type 14 is input only if "ILOD" = 1 or 2 on card 1. Node which loads input on this card will be applied to.
	UX (J)	F (12-24)	Load in X direction on node J
	UY (J)	F (25-36)	Load in Y direction on node J
	UZ (J)	F (37-48)	Load in Z direction on node J

A card Type 14 is input for each node being loaded. Input of Card Type 14 is terminated by a card with J = 300.

CARD TYPE	VARIABLE	FIELD	COMMENT
15			Card Type 15 is input only if "ILOD" = 2 or 3 on Card 1.
	J	I (1 - 4)	Node for which fixity is read in.
	Code (J)	F (13-24)	Fixity of node J. Convention shown below is used.

Code (J)	Freedoms Fixed
0.0	None
1.0	X
2.0	Y
3.0	Z
4.0	X, Y
5.0	X, Z
6.0	Y, Z
7.0	X, Y, Z

Card Formats

Input to the Tetra-core analysis program must be in the form of data cards punched according to fixed formats.

Integer Fields: I (1-4), I (26-30), etc.

An integer must be right adjusted in the field. Unneeded field space may be left blank. Blanks are interpreted as zeros in the corresponding column. Decimal points are not used.

Decimal Fields: F(1-12), F(37-48), etc.

A decimal number, punched with a decimal point, may be located anywhere within the field. Blank fields are interpreted as zeros.

Alphanumeric Fields: A(1-27), A(21-30), etc.

Alphanumeric characters may be anywhere in the field. Legal characters for this field include all FORTRAN characters.

LIST OF SUBROUTINES

OVERLAY TETRA 1 - ROOT OVERLAY

MAIN - Main program. Calls subroutines GEN, STRSS, SOLPAC, DEFL.

OUTIN - Used to read and write stiffness matrix and load vectors on file 4. Called by STIFF, SOLPAC, DEFL.

OVERLAY TETRA 2

GEN - Subroutine to read and print input data. Calls subroutines NODGEN, PLATEN, LODGEN, NODGAN, PLATGA, LODGAN, OPTM, NONLIN. Called by MAIN.

CURV - Curve fitting subroutine, used in generation of airfoil shape. Called by NODGEN, NODGAN.

OPTM - Optimization subroutine. Determines direction of travel to obtain minimum weight design. Calls WEIGHT, DAR. Called by GEN.

WEIGHT - Subroutine to calculate weight of Tetra-core model being optimized. Calls NODGEN, NODGAN. Called by OPTM.

DAR - Subroutine to calculate direction cosines of gradient given partial derivatives. Called by OPTM.

NONLIN - Subroutine to increment loads before each step of nonlinear analysis. Called by GEN.

OVERLAY TETRA 3

NODGEN - Subroutine to generate node points for true Tetra-core model. Calls CURV.

OVERLAY TETRA 4

PLATGN - Subroutine to generate plates for true Tetra-core model. Called by GEN.

OVERLAY TETRA 5

LODGEN - Subroutine to generate nodal loads and nodal fixities for true Tetra-core model. Called by GEN.

OVERLAY TETRA 6

NODGAN - Subroutine to generate node points for truncated Tetra-core model. Called by GEN, WEIGHT.

OVERLAY TETRA 7

PLATGA - Subroutine to generate plates for truncated Tetra-core model. Called by GEN.

OVERLAY TETRA 8

LODGAN - Subroutine to generate nodal loads and nodal fixities for truncated Tetra-core model. Called by GEN.

OVERLAY TETRA 9

STRSS - Overlay to perform finite-element stress analysis on model generated by previous overlays. Called by MAIN. Calls STRESS, STIFF.

OVERLAY TETRA 10

STRESS - Subroutine to renumber Tetra-core model to account for midside nodes. Called by STRSS.

NID - Function to generate new number for midpoint node.

OVERLAY TETRA 11

STIFF - Subroutine to merge element stiffness matrices into overall stiffness matrix. Called by STRSS. Calls TRPRD, TRIM6, OUTIN.

TRPRD - Subroutine to perform matrix multiplication $[A]^T[B][A]$. Called by TRIM6.

TRIM6 - Subroutine to generate linear strain triangle and quadrilateral and constant strain triangular stiffness and stress matrices. Called by STIFF. Calls TRPRD.

OVERLAY TETRA 12

SOLPAC - Subroutine for solution of stiffness matrix. Solves Eqn $P = K\Delta$ using Choleski triangularization and back substitution. Called by MAIN.

OVERLAY TETRA 13

DEFL - Subroutine for printing of deflections, calculation of plate element stresses. Called by MAIN. Calls MARGIN, MODUL, BUKL, PLST.

MARGIN - Subroutine to calculate margin of safety in each plate element using von Mises-Tsai failure criterion. Called by DEFL.

MODUL - Subroutine to calculate plate secant moduli based on strains within the plate. Called by DEFL.

BUKL - Subroutine to calculate buckling stress for typical plate from each leg. Called by DEFL.

PLST - Function to calculate stress given the strain in a plate.

DESCRIPTION OF VARIABLES USED IN COMMON

COMMON/COMN/

NUMNP - Number of nodal points in model

NUMEL - Number of plate elements in model

NLOD - Number of load cases

NLIN - Used to control noninear analysis

MBAND - 1/2 bandwidth of stiffness matrix, in freedoms

NBAND - 1/2 bandwidth of stiffness matrix, in nodes

COMMON/MATL/

E - Moduli of elasticity for each leg

FXT, FYT, FXC, FYC, SSS - Allowable stresses of material in each leg

CMS - Margin of safety in each leg

COMMON/LODA/

XLOAD, YLOAD, XYLOD, XMOM, YMOM, XYMOM, TORQ, XQHR, YQSHR - input applied loads

COMMON/ONE/

SIDE, HT - Tetrahedron side length and actual height (see Figure 4)

THK - Thickness of each leg

THETA, Y1OFF, Y2OFF - Parameters to control variation from theoretical equalateral tetrahedron (see Figure 4)

THT - Theoretical height at which tetrahedron sides would meet (see Figure 4)

DUM - Not used

TC, XC - Thickness/chord and X/chord ratios used to generate airfoil model

HED - Title

NX, NY - Number of legs on X and Y face. Called LX and LY on input sheets (see Figure 5)

IOPT - Used to control optimization

ITYP - Type of model to be run--flat plate, cylinder, airfoil

IFACE - Control of face sheet generation

IHOLE - Used to control if hole will be placed in model or not

ITET - Used to control type of tetrahedron generated--true or truncated

ICNT - Counter for number of steps that have been run in nonlinear or optimization analysis

NNEW - Used to determine which leg a given plate is in

COMMON/OPTA/

NSTP - Limiting number of steps for nonlinear analysis. limiting number of cycles for optimization

IPRT - Controls printing of intermediate printout in optimization

IDS - Controls steepest descent mode in optimization

IDRV - Controls calculation of derivatives in optimization

ITOT - Counter for number of cycles in optimization

ICYC - Counter for number of sidesteps in optimization

LAST - Not used

IK - Used to identify variable being incremented when calculating derivatives in optimization

WTL - Base weight before start of calculation of derivatives in optimization

XMSL - Margin of safety at start of last sidestep in optimization

XMS - Present margin of safety

WTMN - Minimum weight found in optimization

TMLT - Step length used in sidestepping - optimization

WT - Present weight

HMAX - Value of variables associated with WTMN - optimization

HL - Value of variables at start of calculation of derivatives - optimization

CMSA - Value of margins of safety at start of calculation of derivatives - optimization

SLW - Partial derivatives of effect of variables on weight - optimization

SLOPE - Partial derivatives of effect of variables on margin of safety - optimization

RHO - Density of material in each leg

PWT - Gradient of constant weight surface - optimization

PMT - Composite constraint gradient - in optimization

ISA, ISB, ISC - Used to control variables during optimization

COMMON/MODU/

SIGT1, SIGT2, SIGC1, SIGC2, SIGSS - Stresses defining stress-strain curves used in nonlinear analysis

PRXT, PRXC - Poisson's ratios used in nonlinear analysis

STT1, STT2, STC2, STSS - Strain increments used to define stress-strain curves for nonlinear analysis

COMMON/LODB/

APPLD - XLOAD, YLOAD, etc. for each load case are stored in APPLD

COMMON/RAX/

NR4 - Pointer for next record to be read on File 4

NR4LOD - Gives location of first record in loads vector on File 4

NR4STF - Gives location of first record in stiffness matrix on File 4

NR4DIA - Gives location of first record in diagonal on File 4

NREC - Number of records required to store three rows of stiffness matrix on File 4

NRECST - Number of records in stiffness matrix

NRECLD - Number of records in loads vector

NR1 - Pointer for next record to be read on File 1

NR2 - Pointer for next record to be read on File 2

DESCRIPTION OF VARIABLES USED IN SUBROUTINES

SUBROUTINES - GEN, NODGEN, PLATGN, LODGEN, NODGAN, PLATGA, LODGAN, WEIGHT

X, Y, Z - Nodal coordinates

SLOPA - Slope of surface at each node--used for airfoil

NA, NB, NC, ND - Nodes at vertices of plate element

CODE - Fixity for each node

UX, UY, UZ - Loads on each node

SUBROUTINE - OPTM

SLP - Intermediate step in calculation of composite constraint gradient

PMS - Gradient to each margin of safety surface

H - Vector of variables to be optimized

SUBROUTINE - STRESS, STIFF, TRIM6

X,Y,Z,UX,UY,UZ,CODE - See subroutines NODGEN, PLATGEN, etc.

IDO - Node numbers as renumbered to include mid-side nodes

ID - Node numbers in original order, not including midside nodes

NI - Node number of new mid-side node

IN - Sequence of new mid-side node

ML - Added mid-side nodes in plate element

LM - Original vertex nodes inplate element - in original nodal numbering sequence

MM - Renumbered mid-side and vertex nodes in plate element

DUM1,DUM2,DUM3,DUM4 - Dummy variables used in renumbering of load vector and nodal fixities to include the effect of added midside nodes

S - Element stiffness matrix

ETR,ELGT,EPRM, G - Plate elasticity matrix - ETR = $EY/(1-\mu_{xy}\mu_{yx})$, ELGT = $Ex/(1-\mu_{xy}\mu_{yx})$, EPRM = $Ey\mu xy/(1-\mu xy\mu yx)$, G = GXY

T - Plate thickness

ST - Plate stress matrix

IMTL - Indicates leg that plate is in

A - Three rows of merged stiffness matrix

DIAG - Diagonal of merged stiffness matrix

PHYX, PHYY - Used in computing element stiffness matrix

ALAM - Element transformation matrix. Transforms stiffnesses from local to global coordinates

AL, AL2 - Direction cosines of element

QMAT - Elasticity matrix

AST, AK21, DUM, STA, STR, STRA - Dummy matrices used in calculating stiffness and stress matrices

ZETA - Area coordinates of location in triangle for which stresses are calculated

IK - Order in which triangle stiffnesses are added into quadrilateral stiffness matrix

SUBROUTINE SOLPAC

DIAG - Diagonal of stiffness matrix

AI - Three rows of stiffness matrix

AJ - Three columns of stiffness matrix

B - Initial loads vector, deflections after solution is complete

X - Loads vector after reduction

SUBROUTINES DEFL, BUKL, MODUL

P - Displacements for nodes connecting to an element

LM, MM, ST, IDO - See subroutine STRESS

STRESS - Element stresses

FBKL - Buckling stress of an element in each leg

STRAIN - Element strains

B - Nodal deflections

D - Plate stiffnesses

XL - Width of plate

```
//JC710280 JOB ,ST031100060C
IA55I  111806
// ALLOC ONE,091='SYSWK1',300,FMT
// LABEL 120
/*
/&
   111832
```

Nov. 17, 1971

```
//JC710280 JOB ,ST031100060C
IA55I  111926
// ALLOC TWO,091='SYSWK1',300,FMT
// LABEL 60
/*
/&
   111951
```

Nov. 17, 1971

```
//JC710280 JOB ,ST031100060C
IA55I  113800
// ALLOC FOUR,091='SYSWK1',300,FMT
// LABEL 2922
/*
/&
   113844
```

Nov. 17, 1971

```
//JC710280 JOB ,ST031100060C
IA55I  113857
// EXEC UTILS
```

Nov. 17, 1971

```
MAP DVADR=091,VOLID='SYSWK1'

MAP OF 'SYSWK1'
```

DATA SET NAME	SEQ	TYPE	LOW EXTENT	HIGH EXTENT	EXPIRED	LAST REC. WRITTEN
ONE	0	01	CC 000 HH 1	CC 019 HH 0		CC 019 HH 0 R 08
TWO	0	01	CC 019 HH 1	CC 038 HH 0		CC 038 HH 0 R 08
FOUR	0	01	CC 038 HH 1	CC 188 HH 0		CC 188 HH 0 R 01
VTOC	0	81	CC 199 HH 0	CC 199 HH 1		

```
//JC710280 JOB ,ST031100060C
IA55I  141024
//STRESS EXEC FORTRAN(BCD,MAP)

//JC710280 JOB ,ST031100060C
IA55I  141139
//TETMN  EXEC FORTRAN(BCD,MAP)
```

```
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0001

C     ****************************** ROOT OVERLAY ********************************
C     ****************************** ROOT OVERLAY ********************************
      COMMON / COMN / NJMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
      COMMON / MATL / E(4,5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
      COMMON / LODA / XLOAD,YLOAD,XYLOD,XMOM,YMOM,XYMOM,TORQ,XQSHR,YQSHR
      COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
     1  ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
      COMMON / OPTA / NSTP,IPRT,IDS,IDRV,ITOT,ICYC,LAST,IU,   WTL,XMSL
     1  ,XMS,WTMN,TMLT,WT,       HMAX(11),HL(11),CMSA(11)
     2  ,SLW(11),SLCPE(11,5 ),RHO(5), PWT(11),PMT(11)
     3  ,ISA(11),ISB(11),ISC(11)
      COMMON / MODU / SIGT1( 5,9),SIGT2( 5,9),SIGC1( 5,9),SIGC2( 5,9)
     1  ,SIGSS( 5,9),PRXT( 5,9),PRXC( 5,9),STT1( 5),STT2( 5),STC1( 5)
     2  ,STC2( 5),STSS( 5)
      COMMON /LODB / APPLD(9,10)
      COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLD,NR1,NR2
      COMMON /DJMP / ITEST
      DEFINE FILE 1(300,120,U,NR1)
      DEFINE FILE 2(300,60,U,NR2)
      DEFINE FILE 4(300,2922,U,NR4)
      NREC=890
   50 IU = 0
      ICYC = 0
      IDRV = 0
      ICNT = 0
      ITOT = 0
      IOPT = 0
      IDS = 1
      TMLT = 1.0
      WTMN = 1000.
      THK(4) = 0.0
      THK(5) = 0.0
      DUM = 0.0
      DO 100 I = 1,11
      ISA(I) = 0
      ISB(I) = 0
      ISC(I) = 0
  100 SLW(I) = 1.0
   10 CONTINUE
      CALL LOAD('TETRA1')
      CALL GEN
      CALL LOAD('TETRA8')
      CALL STRSS
      CALL LOAD('TETRA11')
      CALL SOLPAC
      CALL LOAD('TETRA12')
      CALL DEFL
      IF (IOPT .EQ. 0 .AND. NLIN.EQ.0) GO TO 50
      IF (IOPT .EQ. 5) GO TO 50
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0002

      GO TO 10
      END
```

```
                                   COMMON BLOCK / CONN   / MAP SIZE  00006C
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
NUMNP     000000       NUMEL     000004       NLOD      000008       NLIN      00000C       MBAND     000010
NUMBLK    000014       NBN       000018       NEL       00001C

                                   COMMON BLOCK / MATL   / MAP SIZE  0000DC
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
E         000000       FXT       000050       FYT       000064       FXC       000078       FYC       00008C
SSS       0000A0       CMS       0000B4

                                   COMMON BLOCK / LODA   / MAP SIZE  000024
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
XLOAD     000000       YLOAD     000004       XYLOD     000008       XMOM      00000C       YMOM      000010
XYMOM     000014       TORQ      000018       XQSHR     00001C       YQSHR     000020

                                   COMMON BLOCK / ONE    / MAP SIZE  0000E8
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
SIDE      000000       HT        000004       THK       000008       THETA     00001C       YIOFF     000020
Y2OFF     000024       THT       000028       DUM       00002C       TC        000030       XC        00005C
HED       000088       NX        0000B8       NY        00008C       IOPT      0000C0       ITYP      0000C4
IFACE     0000C8       IHOLE     0000CC       ITET      0000D0       ICNT      0000D4       NNEW      0000D8

                                   COMMON BLOCK / OPTA   / MAP SIZE  000284
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
NSTP      000000       IPRT      000004       IDS       000008       IDRV      00000C       ITOT      000010
ICYC      000014       LAST      000018       IU        00001C       WTL       000020       XMSL      000024
XMS       000028       WTMN      00002C       TMLT      000030       WT        000034       HMAX      000038
HL        000054       CMSA      000090       SLW       00008C       SLOPE     0000E8       RHO       0001C4
PWT       0001D8       PMT       000204       ISA       000230       ISB       00025C       ISC       000288

                                   COMMON BLOCK / MODU   / MAP SIZE  000550
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
SIGT1     000000       SIGT2     000084       SIGC1     000168       SIGC2     00021C       SIGSS     0002D0
PRXT      000384       PRXC      000438       STT1      0004EC       STT2      000500       STC1      000514
STC2      000528       STSS      00053C

                                   COMMON BLOCK / LODB   / MAP SIZE  000168
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
APPLD     000000

                                   COMMON BLOCK / RAX    / MAP SIZE  000024
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
NR4       000000       NR4LOD    000004       NR4STF    000008       NR4DIA    00000C       NREC      000010
NRECST    000014       NRECLD    000018       NR1       00001C       NR2       000020

                                   COMMON BLOCK / DUMP   / MAP SIZE  000004
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                              PAGE  0004
ITEST      000000

                                         SCALAR MAP
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
I        000108

                                       SUBPROGRAMS CALLED
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
DIOCS#   00010C       LOAD     000110       GEN      000114       STRSS    000118       SOLPAC   00011C
DEFL     000120       IBCOM#   000124

                                         LABEL MAP
LABEL    LOCATION     LABEL    LOCATION     LABEL    LOCATION     LABEL    LOCATION     LABEL    LOCATION
    50   0001CC          100   000268          10    000282

 TOTAL MEMORY REQUIREMENTS 000358 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//OUTIN  EXEC FORTRAN(BCD,MAP)

FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                              PAGE  0001

                 SUBROUTINE OUTIN(N,A,M)
                 COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLD,NR1,NR2
                 DIMENSION A(M)
                 IF(N.EQ.1) WRITE( 4'NR4) A
                 IF(N.EQ.2)  READ( 4'NR4) A
                 RETURN
                 END

FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                              PAGE  0002

                                   COMMON BLOCK / RAX     / MAP SIZE  000024
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
NR4      000000       NR4LOD   000004       NR4STF   000008       NR4DIA   00000C       NREC     000010
NRECST   000014       NRECLD   000018       NR1      00001C       NR2      000020

                                         SCALAR MAP
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
M        0000DC       N        0000E0

                                         ARRAY MAP
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
A        0000E4

                                       SUBPROGRAMS CALLED
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
IBCOM#   0000E8

 TOTAL MEMORY REQUIREMENTS 00026C BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//GEN    EXEC FORTRAN(BCD,MAP)
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4    DATE  71321                        PAGE  0001

               C        ************************ OVERLAY A ****************************************
 0001                   SUBROUTINE GEN
 0002                   COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
 0003                   COMMON / MATL / E(4,5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
 0004                   COMMON / LODA / XLOAD,YLOAD,XYLOD,XMOM,YMOM,XYMOM,TORQ,XQSHR,YQSHR
 0005                   COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
                      1  ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
 0006                   COMMON / OPTA / NSTP,IPRT,IDS,IDRV,ITOT,ICYC,LAST,IU,  WTL,XMSL
                      1  ,XMS,WTMN,TMLT,WT,      HMAX(11),HL(11),CMSA(11)
                      2  ,SLW(11),SLOPE(11,5 ),RHO(5), PWT(11),PMT(11)
                      3  ,ISA(11),ISB(11),ISC(11)
 0007                   COMMON / MODU / SIGT1( 5,9),SIGT2( 5,9),SIGC1( 5,9),SIGC2( 5,9)
                      1  ,SIGSS( 5,9),PRXT( 5,9),PRXC( 5,9),STT1( 5),STT2( 5),STC1( 5)
                      2  ,STC2( 5),STSS( 5)
 0008                   COMMON /LODB / APPLD(9,10)
 0009                   DIMENSION X(300),Y(300),Z(300),SLOPA(300)
 0010                   COMMON /DUMP / ITEST
 0011                   REWIND 3                                                          TAPE 3
 0012                   IF (ICNT .GT. 0) GO TO 210
 0013                   DO 50 I = 1,11
 0014                   TC(I) = 0.0
 0015              50 XC(I) = 0.0
 0016                   READ (5,5020) NLOD,IOPT,NLIN,ITYP,IFACE,IHOLE,NSTP,ITET,ITEST
 0017            5020 FORMAT (9I4)
 0018                   IF (NLOD .EQ. 0) CALL EXIT
 0019                   IF (NSTP .EQ. 0) NSTP = 3
 0020                   WRITE(6,6010) NLOD,IOPT,NLIN,ITYP,IFACE,IHOLE,NSTP,ITET
 0021            6010 FORMAT ( 1H1, //,7X,  5HLOADS, 5X, 3HOPT, 5X, 7HPLASTIC, 5X,
                      1    4HTYPE, 5X, 4HFACE, 5X, 4HHOLE, 5X, 5HSTEPS,5X,5HTETRA,/,8I10)
 0022                   IF (ITYP .EQ. 1) WRITE (6,6110)
 0023                   IF (ITYP .EQ. 2) WRITE (6,6120)
 0024                   IF (ITYP .EQ. 3) WRITE (6,6130)
 0025            6110 FORMAT ( /// , 5X, 31HTETRA-CORE FLAT PLATE ANALYSIS  )
 0026            6120 FORMAT ( /// , 5X, 31HTETRA-CORE CYLINDER ANALYSIS      )
 0027            6130 FORMAT ( /// , 5X, 31HTETRA-CORE AIRFOIL ANALYSIS       )
 0028                   N = 3
 0029                   IF(IFACE .EQ. 1) N = 5
 0030                   JK = N + 2
 0031                   IF (IOPT .GT. 0) READ (5,5025) (ISA(I),ISB(I),ISC(I),I=1,11)
 0032            5025 FORMAT ( 3I4 )
 0033                   IF (IOPT .GT. 0) WRITE (6,6015) (ISA(I),ISB(I),ISC(I),I=1,11)
 0034            6015 FORMAT ( /// , 5X, 20HOPTIMIZATION OPTIONS / 5X, 3HISA,
                      1     5X, 3HISB, 5X, 3HISC , /, (3I8,/) )
 0035             100 READ (5,5010) HED
 0036            5010 FORMAT (12A4)
 0037                   WRITE(6,5010) HED
 0038                   READ (5,5030) (E(1,I),E(2,I),E(3,I),E(4,I),I=1,N)
 0039            5030 FORMAT ( 4F12.5)
 0040                   WRITE(6,6020) (E(1,I),E(2,I),E(3,I),E(4,I),I=1,N)
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4    DATE  71321                        PAGE  0002

 0041            6020 FORMAT ( /// , 5X, 2HEX,10X, 2HEY, 10X, 3HGXY,  8X, 4HMUXY,
                      1  / ( 3F12.0,F12.5) )
 0042                   READ (5,5040) NX,NY,SIDE,HT ,(THK(I),I=1,N)
 0043            5040 FORMAT (2I4,4X,8F6.0)
 0044                   IF (HT .LT. .001) HT = .81649*SIDE
 0045                   WRITE(6,6030) NX,NY,SIDE,HT, (THK(I),I=1,N)
```

```
0046          6030 FORMAT ( ///, 8X, 2HLX, 8X, 2HLY, 5X, 4HSIDE, 5X, 6HHEIGHT ,
                  1 12X, 15HLEG THICKNESSES, /, 2I10,8F12.5 )
0047               READ (5,5035) THETA,Y1OFF,Y2OFF,THT
0048               IF (THT .LT. HT) THT = HT
0049               WRITE (6,6035) THETA,Y1OFF,Y2OFF,THT
0050          6035 FORMAT ( /// , 5X, 5HTHETA, 7X, 5HY1OFF, 7X, 5HY2OFF, 8X,
                  1  3HTHT, / , 4F12.4  )
0051               IF (ITYP .EQ. 3) READ (5,5045) (TC(I),XC(I),I=1,11)
0052               IF (ITYP .EQ. 3) WRITE(6,5045) (TC(I),XC(I),I=1,11)
0053          5045 FORMAT ( 2F12.5 )
0054               IF (NLIN .NE. 1) GO TO 120
0055               DO 110 IK = 1,N
0056               READ (5,5050)  STT1(IK),STT2(IK),STC1(IK),STC2(IK),STSS(IK)
0057               READ (5,5050)  (SIGT1(IK,J),J=1,7)
0058               READ (5,5050)  (SIGT2(IK,J),J=1,7)
0059               READ (5,5050)  (SIGC1(IK,J),J=1,7)
0060               READ (5,5050)  (SIGC2(IK,J),J=1,7)
0061               READ (5,5050)  (SIGSS(IK,J),J=1,7)
0062               READ (5,5050)  (PRXT (IK,J),J=1,7)
0063               READ (5,5050)  (PRXC (IK,J),J=1,7)
0064           110 CONTINUE
0065           120 READ (5,5035) (FXT(I),FYT(I),FXC(I),FYC(I),SSS(I),I=1,N)
0066          5035 FORMAT ( 5F12.3)
0067               WRITE(6,6070) (FXT(I),FYT(I),FXC(I),FYC(I),SSS(I),I=1,N)
0068          6070 FORMAT ( ///, 20X,  18HALLOWABLE STRESSES /, 9X, 3HFXT, 9X, 3HFYT,
                  1  9X,3HFXC, 9X, 3HFYC, 9X, 3HFXY, /  (5F12.0))
0069               IF (IOPT .NE. 1) GO TO 145
0070               READ (5,5035) (RHO(I),I=1,N)
0071               WRITE(6,6075) (RHO(I),I=1,N)
0072          6075 FORMAT ( //, 5X, 13HLEG DENSITY # , 5F12.5 )
0073           145 IF (ITET) 150,150,160
0074          150  CALL LOAD('TETRA2')
0075               CALL NODGEN (O,X,Y,Z,SLOPA)
0076          155  CALL LOAD('TETRA3')
0077               CALL PLATGN
0078               GO TO 170
0079          160  CALL LOAD('TETRA5')
0080               CALL NODGAN (O,X,Y,Z,SLOPA)
0081               CALL LOAD('TETRA6')
0082               CALL PLATGA
0083           170 DO 200 N = 1,NLOD
0084               READ (5,5050) XLOAD,YLOAD,XYLOD ,XMOM,YMOM,XYMOM,TORQ,XQSHR,YQSHR
0085          5050 FORMAT ( 6F12.0 , F8.0 )
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4    DATE  71321                          PAGE  0003

0086               WRITE(6,6080) N,XLOAD,YLOAD,XYLOD,XMOM,YMOM,XYMOM,TORQ,XQSHR,YQSHR
0087          6080 FORMAT ( /// ,20X, 9HLOAD CASE , I4,/, 9X,2HNX, 10X, 2HNY, 9X,
                  1 3HNXY, 10X, 2HMX, 10X,2HMY, 10X, 3HMXY, 6X, 6HTORQUE, 8X,
                  1  2HQX, 10X, 2HQY, / , 9F12.0 )
0088               APPLD(1,N) = XLOAD
0089               APPLD(2,N) = YLOAD
0090               APPLD(3,N) = XYLOD
0091               APPLD(4,N) = XMOM
0092               APPLD(5,N) = YMOM
0093               APPLD(6,N) = XYMOM
0094               APPLD(7,N) = TORQ
0095               APPLD(8,N) = XQSHR
0096               APPLD(9,N) = YQSHR
0097               IF (ITET) 180,180,190
0098          180  CALL LOAD('TETRA4')
```

```
      CALL LODGEN (X,Y,Z,SLOPA)
      GO TO 200
190   CALL LOAD('TETRA7')
      CALL LODGAN (X,Y,Z,SLOPA)
200   CONTINUE
      ICNT = 1
      GO TO 250
210   CONTINUE
      IF (IOPT .GT. 0) CALL OPTM (X,Y,Z,SLOPA)
      IF (NLIN .EQ. 1) CALL NONLIN (ICNT,IOPT,NSTP)
      IF (ITET) 220,220,230
220   CALL LOAD('TETRA2')
      CALL NODGEN (O,X,Y,Z,SLOPA)
      CALL LOAD('TETRA3')
      CALL PLATGN
      CALL LOAD('TETRA4')
      CALL LODGEN (X,Y,Z,SLOPA)
      RETURN
230   CALL LOAD('TETRA5')
      CALL NODGAN (O,X,Y,Z,SLOPA)
      CALL LOAD('TETRA6')
      CALL PLATGA
      CALL LOAD('TETRA7')
      CALL LODGAN (X,Y,Z,SLOPA)
250   RETURN
      END
```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0004

COMMON BLOCK / COMN / MAP SIZE 00006C

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	000000	NUMEL	000004	NLOD	000008	NLIN	00000C	MBAND	000010
NUMBLK	000014	NBN	000018	NEL	00001C				

COMMON BLOCK / MATL / MAP SIZE 00000C

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
E	000000	FXT	000050	FYT	000064	FXC	000078	FYC	00008C
SSS	0000A0	CMS	0000B4						

COMMON BLOCK / LODA / MAP SIZE 000024

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
XLOAD	000000	YLOAD	000004	XYLOD	000008	XMOM	00000C	YMOM	000010
XYMOM	000014	TORQ	000018	XQSHR	00001C	YQSHR	000020		

COMMON BLOCK / ONE / MAP SIZE 0000E8

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	000000	HT	000004	THK	000008	THETA	00001C	Y1OFF	000020
Y2OFF	000024	THT	000028	DUM	00002C	TC	000030	XC	00005C
HED	000088	NX	000098	NY	0000BC	IOPT	0000C0	ITYP	0000C4
IFACE	0000C8	IHOLE	0000CC	ITET	0000D0	ICNT	0000D4	NNEW	0000D8

COMMON BLOCK / OPTA / MAP SIZE 000284

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NSTP	000000	IPRT	000004	IDS	000008	IDPV	00000C	ITOT	000010
ICYC	000014	LAST	000018	IU	00001C	WTL	000020	XMSL	000024
XMS	000028	WTMN	00002C	TMLT	000030	WT	000034	HMAX	000038
HL	000064	CMSA	000090	SLW	0000BC	SLOPE	0000E8	RHO	0001C4
PMT	0001D8	PMT	000204	ISA	000230	ISB	00025C	ISC	000288

```
                                            COMMON BLOCK / MODU    / MAP SIZE  000550
SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION
SIGT1    000000      SIGT2    000084      SIGC1    000168      SIGC2    00021C      SIGSS    0002D0
PRXT     000384      PRXC     000438      STT1     0004EC      STT2     000500      STC1     000514
STC2     000528      STSS     00053C

                                            COMMON BLOCK / LODB    / MAP SIZE  000168
SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION
APPLD    000000

                                            COMMON BLOCK / DUMP    / MAP SIZE  000004
SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION
ITEST    000000

                                            SCALAR MAP
SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION
I        00020C      N        000210      JK       000214      IK       000218      J        00021C
  FORTRAN IV    MODEL 44   PS          VERSION 3,  LEVEL 4   DATE  71321                     PAGE  0005
                                            ARRAY MAP
SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION
X        000220      Y        0006D0      Z        000B80      SLOPA    001030

                                            SUBPROGRAMS CALLED
SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION    SYMBOL   LOCATION
IBCOM#   0014E0      EXIT     0014E4      LOAD     0014E8      NODGEN   0014EC      PLATGN   0014F0
NODGAN   0014F4      PLATGA   0014F8      LODGEN   0014FC      LODGAN   001500      OPTN     001504
NONLIN   001508

                                            LABEL MAP
LABEL    LOCATION    LABEL    LOCATION    LABEL    LOCATION    LABEL    LOCATION    LABEL    LOCATION
   50    001632       5020    001760       6010    001814       6110    0018F8       6120    001926
 6130    001954       5025    001A28       6015    001AA8        100    001AEA       5010    001B10
 5030    001BA8       6020    001C20       5040    001CC8       6030    001D6C       6035    001E40
 5045    001F44        110    002248        120    00225A       5035    0022CC       6070    002348
 6075    002440        145    002462        150    002472        155    00248A        160    0024A0
  170    0024C8       5050    00253C       6080    002584        180    002680        190    00269E
  200    002686        210    0026E2        220    00273A        230    00277E        250    0027BA
  TOTAL MEMORY REQUIREMENTS 00281C BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//CURV    EXEC FORTRAN(BCD,MAP)

                SUBROUTINE CURV (X,Y,A,B,C)
                DIMENSION A(11),B(11)
                Y = 1000000.
                C = 0.0
                DO 100 I = 1,11
                IF (X - A(I)) 110,100,100
            100 CONTINUE
                GO TO 120
            110 CONTINUE
                J = I - 1
                C = (B(I)-B(J)) / (A(I)-A(J))
                Y = B(J) + (X-A(J)) * C
                C = ATAN2 ((B(I)-B(J)),(A(I)-A(J)))
            120 CONTINUE
                RETURN
                END
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4    DATE  71321                         PAGE  0002

                                           SCALAR MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
Y         0000E0      C         0000E4      I         0000E8      X         0000EC      J         0000F0

                                           ARRAY MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
A         0000F4      B         0000F8

                                       SUBPROGRAMS CALLED
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
ATAN2     0000FC

                                           LABEL MAP
LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION
  100     0001F6        110     000212        120     000292

 TOTAL MEMORY REQUIREMENTS 0002E0 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//OPTIM  EXEC FORTRAN(BCD,MAP)

FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4    DATE  71321                         PAGE  0001

0001               SUBROUTINE OPTM(X,Y,Z,SLOPA)
0002               COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBV,VEL(20)
0003               COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
                 1  ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
0004               COMMON / MATL / E(4,5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
0005               COMMON / OPTA / NSTP,IPRT,IDS,IDRV,ITOT,ICYC,LAST,IK,  WTL,XMSL
                 1  ,XMS,WTMN,TMLT,WT,        HMAX(11),HL(11),CMSA(11)
                 2  ,SLW(11),SLOPE(11,5 ),RHO(5), PWT(11),PMT(11)
                 3  ,ISA(11),ISB(11),ISC(11)
0006               DIMENSION SLP(11),PMS(11,5),H(12)
0007               DIMENSION X(300),Y(300),Z(300),SLOPA(300)
0008               EQUIVALENCE (H,SIDE)
0009               IPRT = 1
0010          6000 FORMAT ( 5E20.5 )
0011          6020 FORMAT ( 10F12.5 )
0012               M = 3
0013               IF (IFACE .EQ. 1) M = 5
0014               N = 11
0015               IF (ICNT .EQ. 1 .AND. IPRT .EQ. 1) WRITE (6,6005)
0016          6005 FORMAT ( 1H1,//,5X,18HOPTIMIZATION STEPS  ///  )
0017               ICNT = ICNT + 1
0018               IF (ICNT .GT. 200) IOPT = 5
0019               CALL WEIGHT (WT,RHO,NUMEL,X,Y,Z,SLOPA)
0020               XMS = 10.
0021               DO 120 I = 1,M
0022           120 IF (XMS .GT. CMS(I)) XMS = CMS(I)
0023           130 CONTINUE
0024               IF (IPRT - 1) 150,140,150
0025           140 WRITE (6,6100) IK,ICYC,ICNT,ITOT
0026          6100 FORMAT ( /, 5X, 4HIK #, I4, 5X, 11HSIDESTEPS #, I4, 5X, THSTEPS #,
                 1  I4 , 5X,  8HCYCLES # , I4  )
0027               WRITE (6,6110) WT,XMS,(H(I),I=1,11)
```

```
0028          6110 FORMAT (  5X, 4HWT #, F12.5, 5X, 8HMARGIN #, F12.5, 5X
                  1 , 6HHMIK< #, 5F12.5 / 7F12.5)
0029               WRITE (6,6120) (CMS(K),K=1,M)
0030          6120 FORMAT (  5X, 11HMARGINSi< #, 5F12.5  )
0031           150 IF (IDS - 1) 160,550,160
0032           160 IF (IOPT- 5) 170,590,170
0033           170 IF (IDRV - 1) 180,340,180
             C     OPTIMIZATION
0034           180 ICYC = ICYC + 1
0035               IDRV = 0
0036               IF (ICYC - 10) 190,190,510
0037           190 IF (XMS.LT. XMSL.AND. ICYC .EQ.2) GO TO 320
0038               IF (XMS.LT. XMSL.AND. ICYC .GT. 2) GO TO 510
0039           200 IF (ICYC .GT. 1) XMSL = XMS
0040               DO 210 K = 1,N
0041           210 HL(K) = H(K)
0042               IF(ICYC - 1) 260,220,260
```



```
             C     CALCULATE DIRECTION COSINES OF CONSTANT WEIGHT SURFACE          C
0043           220 CALL DAR (SLW,PWT,N,1,ISA)
0044               CALL DAR (SLOPE,PMS,N,M,ISA)
0045               DO 230 J = 1,N
0046               SLP(J) = 0.
0047               DO 230 K = 1,M
0048               IF (CMS(K) .GT. .9) CMS(K) = .9
0049           230 SLP(J) = SLP(J) +    PMS(J,K)*(1. - CMS(K))**15
             C     CALCULATE DIRECTION COSINES OF RESULTANT CONSTRAINT SURFACE     C
0050               CALL DAR(SLP,PWT,N,1,ISA)
0051               IF (IPRT - 1) 260,240,260
0052           240 WRITE (6,6030)
0053          6030 FORMAT ( // , 14H SIDE STEPPING , /, 5X, 39HDIRECTION COSINES OF C
                  1ONST. WT. SURFACE ,/, 5X, 21H, CONSTRAINT SURFACES ,/, 5X,
                  2  19H, MARGINS OF SAFETY ,/, 5X, 46H, DIR. COSINES OF COMPOSITE CO
                  3NSTRAINT SURFACE     )
0054               WRITE (6,6020) (PWT(K),K=1,N)
0055               DO 250 K = 1,N
0056           250 WRITE (6,6020) (PMS(K,J),J=1,M)
0057               WRITE (6,6020) (CMS(K),K=1,M)
0058               WRITE (6,6020) (PMT(K),K=1,N)
0059           260 DO 270 K = 1,N
0060           270 H(K) = H(K)*(1. + PMT(K)*TMLT)
0061           280 WT = 0.0
0062               CALL WEIGHT (WT,RHO,NUMEL,X,Y,Z,SLOPA)
0063               DWT = WT / WTL - 1.
0064               IF (ABS(DWT) - .01 ) 640,300,300
0065           300 DO 310 K = 1,N
0066               IF (PWT(K)*DWT      .GT. .8) DWT =   .8/PWT(K)
0067           310 H(K) = H(K)*(1.-PWT(K)*DWT)
0068               GO TO 280
             C     IF FIRST SIDESTEP DIDNT WORK REDUCE STEP LENGTH
0069           320 DO 330 K = 1,N
0070           330 H(K) = HL(K)
0071               TMLT = TMLT/4.
0072               XMS = XMSL
0073               GO TO 200
             C     CALCULATE DERIVITIES
0074           340 IF (IK) 380,350,380
0075           350 IF (IPRT .EQ. 1) WRITE (6,6040)
0076          6040 FORMAT ( 31H    CALCULATION OF DERIVITIVES    )
```

```
      ITOT = ITOT + 1
      XMSL = XMS
      WTL = WT
      DO 360 K = 1,M
  360 CMSA(K) = CMS(K)
      TMLT = TMLT/2.
      DO 370 IJ = 1,N
  370 HL(IJ) = H(IJ)
```



```
  380 CONTINUE
      IF (IK) 390,410,390
  390 SLW(IK) = (WT - WTL)  / (H(IK) - HL(IK))
      DO 400 K = 1,M
  400 SLOPE(IK,K) = (CMS(K) - CMSA(K)) / (H(IK) - HL(IK))
      IF (IPRT .EQ. 1) WRITE(6,6050)(SLOPE(IK,K),K=1,M),SLW(IK)
 6050 FORMAT (7F15.5 )
  410 CONTINUE
      IK = IK + 1
      IF (ISA(IK)-12) 415,410,415
  415 DO 420 IJ = 1,N
  420 H(IJ) = HL(IJ)
      IF (IK - N) 430,430,180
  430 H(IK) = H(IK) + .001
      IF (ISA(IK)) 440,450,440
  440 IA = ISA(IK)
      H(IA) = H(IK)
  450 IF (ISB(IK)) 460,470,460
  460 IB = ISB(K)
      H(IB) = H(IK)
  470 IF (ISC(IK)) 480,490,480
  480 IC = ISB(IK)
      H(IC) = H(IK)
  490 IF (IK - N) 500,500,180
  500 ICYC = 0
      GO TO 640
C     RETURN TO PREVIOUS VALUE BEFORE DESCENDING
  510 DO 520 K = 1,N
  520 H(K) = HL(K)
      IF (IPRT .EQ. 1) WRITE (6,6060)
 6060 FORMAT ( 11H DESCENDING )
      IF (ICYC .GT. 9) TMLT = 4.*TMLT
      IDS = 1
      GO TO 640
C     STORE NEW OPTIMUM
  530 WTMN = WT
      IF (IPRT .EQ. 1) WRITE (6,6070)
 6070 FORMAT ( 17H     NEW OPTIMUM  )
      DO 540 K = 1, N
      IF (IPRT .EQ. 1) WRITE (6,6020) H(K)
  540 HMAX(K) = H(K)
      GO TO 160
C     DESCENDING
  550 IF (ABS(XMS) .LT. .050) IDS = 0
      IF (IDS) 570,560,570
  560 IK = 0
      IDRV = 1
      IF (ITOT .GT. NSTP) IOPT = 5
      IF (ICNT .GT. 100) IOPT = 5
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                          PAGE  0004

0131              IF (WT - WTMN) 530,160,160
           C      CALCULATE DIRECTION COSINES OF CONSTANT WEIGHT SURFACE                    C
0132          570 CALL DAR (SLW,PWT,N,1,ISA)
0133              DO 580 K = 1,N
0134              IF (ITOT .GT. 0) GO TO 575
0135              IF (K .LT. 3) GO TO 580
0136              IF (K .GT. 7) GO TO 580
0137          575 IF (PWT(K)*XMS   .GT. .8) XMS  = .8/PWT(K)
0138              H(K) = H(K)*(1.-PWT(K)*XMS)
0139          580 CONTINUE
0140              GO TO 640
0141          590 CONTINUE
0142              LAST = 1
0143              ICNT = 0
           C      IF ON FINAL CYCLE INSERT OPTIMUM VALUES
0144              WRITE (6,6080)
0145         6080 FORMAT(1H1,//,31H    OPTIMIZED LAYER THICKNESSES    ,/ )
0146              DO 600 K = 1,N
0147          600 WRITE (6,6090)K,HMAX(K)
0148         6090 FORMAT ( I5,F15.5)
0149          610 DO 620 K = 1,N
0150          620 H(K) = HMAX(K)
0151          640 CONTINUE
0152              RETURN
0153              END
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                          PAGE  0005


                                  COMMON BLOCK / COMN    / MAP SIZE  00006C
SYMBOL   LOCATION      SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
NUMNP    000000        NUMEL    000004       NLOD     000008       NLIN     00000C       MBAND    000010
NUPBLK   000014        NBN      000018       NEL      00001C

                                  COMMON BLOCK / ONE     / MAP SIZE  0000E8
SYMBOL   LOCATION      SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
SIDE     000000        H        000000       HT       000004       THK      000008       THETA    00001C
Y1OFF    000020        Y2OFF    000024       THT      000028       DUM      00002C       TC       000030
XC       00005C        HED      000088       NX       000088       NY       00008C       IOPT     0000C0
ITYP     0000C4        IFACE    0000C8       IHOLE    0000CC       ITET     0000D0       ICNT     0000D4
NNEW     0000D8

                                  COMMON BLOCK / MATL    / MAP SIZE  0000DC
SYMBOL   LOCATION      SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
F        000000        FXT      000050       FYT      000064       FXC      000078       FYC      00008C
SSS      0000A0        CMS      0000B4

                                  COMMON BLOCK / OPTA    / MAP SIZE  000284
SYMBOL   LOCATION      SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
NSTP     000000        IPWT     000004       IDS      000008       IDRV     00000C       ITOT     000010
ICYC     000014        LAST     000018       IK       00001C       WTL      000020       XMST     000024
XMS      000028        WTMN     00002C       TMLT     000030       WT       000034       HMAX     000038
HC       000064        CMSA     000090       SLW      0000BC       SLOPE    0000E8       RHO      0001C4
PWT      000108        PMT      000204       ISA      000230       ISB      00025C       ISC      000288

                                  SCALAR MAP
SYMBOL   LOCATION      SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
M        00025C        N        000260       I        000264       K        000268       J        00026C
DWT      000270        IJ       000274       IA       000278       IB       00027C       IC       000280
```

```
                                                ARRAY MAP
SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
SLP        000284        PMS        000280        X          00038C        Y          000390        Z          000394
SLOPA      000398

                                                SUBPROGRAMS CALLED
SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
IBCOM#     00039C        WEIGHT     0003A0        DAR        0003A4        FRXPI#     0003A8

                                                LABEL MAP
LABEL      LOCATION      LABEL      LOCATION      LABEL      LOCATION      LABEL      LOCATION      LABEL      LOCATION
 6000      0004D6         6020      0004E4         6005      000574          120      00060A          130      000646
  140      00065A         6100      00069C         6110      000740         6120      0007C4          150      0007E2
  160      0007FA          170      000812          180      00082A          190      000856          200      0008C2
  210      0008EC          220      00092E          230      000990          240      000A1E         6030      000A38
  250      000B3C          260      000C38          270      000C44          280      000C82          300      000CE2

FORTRAN IV     MODEL 44   PS          VERSION 3,  LEVEL 4    DATE  71321                         PAGE  0006

  310      00001C          320      000062          330      00006E          340      0000BA          350      0000CA
 6040      0000F8          360      000E4E          370      000E90          380      000EC2          390      000ED2
  400      000F10         6050      000FE4          410      000FF2          415      00101E          420      00102A
  430      00106C          440      0010A8          450      00100C          460      0010F4          470      001130
  480      001148          490      00117C          500      001190          510      0011A6          520      0011B2
 6060      001208          530      001256         6070      001288          540      0012F4          550      00132C
  560      00135C          570      0013C6          575      00140E          580      00146A          590      001486
 6080      00148C          600      0014F6         6090      00153E          610      00154C          620      001554
  640      001586

  TOTAL MEMORY REQUIREMENTS 001614 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//WEIGHT EXEC FORTRAN(BCD,MAP)

0001                 SUBROUTINE WEIGHT (WT,RHO,NUMEL,X,Y,Z,SLOPA)
0002                 DIMENSION X(300),Y(300),Z(300),SLOPA(300)
0003                 COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
                  1   ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
0004                 COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLD,NR1,NR2
0005                 DIMENSION AL(3),AL2(3),LM(6),RHO(5)
0006                 IF (ITET) 50,50,60
0007            50   CALL NODGEN (1,X,Y,Z,SLOPA)
0008                 GO TO 70
0009            60   CALL NODGAN (1,X,Y,Z,SLOPA)
0010              70 WT = 0.0
0011                 DO 200 I = 1,NUMEL
0012                 NR2=I
0013                 READ(2'NR2) IDP,IP,IQ,IR,IS,ETR,ELGT,EPRM,G,T,INTL                        TAPE
0014                 IJK = 1
0015                 IF (IS .GT. 0) IJK = 2
0016                 DO 200 K = 1,IJK
0017                 IF (K .EQ. 1) GO TO 71
0018                 IP = IR
0019                 IR = IS
0020              71 XQP = X(IQ)-X(IP)
0021                 YQP = Y(IQ)-Y(IP)
0022                 ZQP = Z(IQ)-Z(IP)
0023                 XRP = X(IR)-X(IP)
0024                 YRP = Y(IR)-Y(IP)
0025                 ZRP = Z(IR)-Z(IP)
0026                 XRQ = X(IR)-X(IQ)
0027                 YRQ = Y(IR)-Y(IQ)
0028                 ZRQ = Z(IR)-Z(IQ)
```

```
          D1= SQRT(XQP*XQP+YQP*YQP+ZQP*ZQP)
      100 AL(1)= XQP/D1
          AL(2)= YQP/D1
          AL(3)= ZQP/D1
      102 RR= AL(1)*XRP +AL(2)*YRP +AL(3)*ZRP
          X2 = XRP -AL(1)*RR
          Y2 = YRP -AL(2)*RR
          Z2 = ZRP -AL(3)*RR
          D2 =SQRT(X2*X2 +Y2*Y2 +Z2*Z2)
          AREA = D1*D2/2.
          WT = WT + AREA*RHO(INTL)*THK(INTL)
      200 CONTINUE
          Y1LGT = .28867 + Y1OFF
          Y2LGT = .86657 + Y2OFF
          XLGT = ((NX-1)*Y2LGT + Y1LGT   )*SIDE
          YLGT = (FLOAT(NY)-.5)*SIDE
          WT = WT/(YLGT*XLGT)
          WRITE (6,6100) WT
     6100 FORMAT (6F 15.5)
          RETURN

FORTRAN IV    MODEL 44   PS        VERSION 3,  LEVEL 4    DATE  71321                   PAGE  0002

              END

FORTRAN IV    MODEL 44   PS        VERSION 3,  LEVEL 4    DATE  71321                   PAGE  0003

                                   COMMON BLOCK / ONE     / MAP SIZE  0000E8
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
SIDE     000000       WT       000004       THK      000008       THETA    00001C       Y1OFF    000020
Y2OFF    000024       THT      000028       DUM      00002C       TC       000030       XC       00005C
WED      000088       NX       000088       NY       0000BC       IOPT     0000C0       ITYP     0000C4
IFACE    0000C8       IHOLE    0000CC       ITET     0000D0       ICNT     0000D4       NNEW     0000D8

                                   COMMON BLOCK / RAX     / MAP SIZE  000024
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
NR4      000000       NR4LOD   000004       NR4STF   000008       NR4DIA   00000C       NREC     000010
NRECST   000014       NRECLD   000018       NR1      00001C       NR2      000020

                                   SCALAR MAP
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
WT       0000FC       I        000100       NUMEL    000104       IOP      000108       IP       00010C
IQ       000110       IR       000114       IS       000118       ETR      00011C       ELGT     000120
EPRM     000124       G        000128       T        00012C       INTL     000130       IJK      000134
K        000138       XQP      00013C       YQP      000140       ZQP      000144       XRP      000148
YRP      00014C       ZRP      000150       XRQ      000154       YRQ      000158       ZRQ      00015C
D1       000160       RR       000164       X2       000168       Y2       00016C       Z2       000170
D2       000174       AREA     000178       Y1LGT    00017C       Y2LGT    000180       XLGT     000184
YLGT     000188

                                   ARRAY MAP
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
X        00018C       Y        000190       Z        000194       SLOPA    000198       AL       00019C
AL2      0001A8       LN       0001B4       RHO      0001CC

                                   SUBPROGRAMS CALLED
SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION     SYMBOL   LOCATION
NODGEN   0001D0       NODGAN   0001D4       IBCOM#   0001D8       SQRT     0001DC
```

```
                                          LABEL MAP
 LABEL    LOCATION      LABEL   LOCATION      LABEL    LOCATION      LABEL    LOCATION      LABEL   LOCATION
    50    0002E2           60   000316           70    000344           71    000424          100   000506
   102    00052A          200   0005E6         6100    0006AC

   TOTAL MEMORY REQUIREMENTS 000710 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//DAR      EXEC FORTRAN(BCD,MAP)

                     SUBROUTINE DAR(A,B,N,M,ISA)
            C        CALCULATE DIRECTION COSINES                                         C
                     DIMENSION A(11,5),B(11,5),ISA(11)
                     DO 30 J = 1,M
                     SUM = 0.
                     DO 10 K = 1,N
                     IF (ISA(K)-12) 5,10,5
                   5 IF (A(K,J) .LT. 1.0 E-10) A(K,J) = 1.0 E-10
                     SUM = SUM + A(K,J) *A(K,J)
                  10 CONTINUE
                     SUM = SUM**.5
                     DO 20 K = 1,N
                     B(K,J) = 0.0
                     IF (ISA(K)-12) 15,20,15
                  15 B(K,J) = A(K,J)/SUM
                  20 CONTINUE
                  30 CONTINUE
                     RETURN
                     END

FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                      PAGE  0002

                                         SCALAR MAP
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
J         0000F0       M         0000F4       SUM       0000F8       K         0000FC       N         000100

                                         ARRAY MAP
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
A         000104       B         000108       ISA       00010C

                                     SUBPROGRAMS CALLED
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
FRXPR#    000110

                                         LABEL MAP
LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION
    5     000206          10     000268          15     0002DA          20     000308          30     00031E

   TOTAL MEMORY REQUIREMENTS 000378 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//NONLIN EXEC FORTRAN(BCD,MAP)

FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                      PAGE  0001

                     SUBROUTINE NONLIN (ICNT,IOPT,NSTP)
                     COMMON / LODA / XLOAD,YLOAD,XYLOD,XMOM,YMOM,XYMOM,TORQ,XQSHR,YQSHR
                     ICNT = ICNT + 1
                     IF (ICNT .GT. NSTP) IOPT = 5
```

```
0005            N = ICNT
0006            FACT = FLOAT(ICNT)/FLOAT(ICNT-1)
0007            XLOAD = XLOAD*FACT
0008            YLOAD = YLOAD*FACT
0009            XYLOD = XYLOD*FACT
0010            XMOM = XMOM*FACT
0011            YMOM = YMOM*FACT
0012            XYMOM = XYMOM*FACT
0013            TORQ = TORQ*FACT
0014            XQSHR = XQSHR*FACT
0015            YQSHR = YQSHR*FACT
0016            WRITE(6,6080) N,XLOAD,YLOAD,XYLOD,XMOM,YMOM,XYMOM,TORQ,XQSHR,YQSHR
0017       6080 FORMAT (1H1,/,20X, 9HLOAD INC. , I4,/, 9X,2HNX, 10X, 2HNY, 9X,
              1 3HNXY, 10X, 2HMX, 10X,2HMY, 10X, 3HMXY, 6X, 6HTORQUE, 8X,
              1  2HQX, 10X, 2HQY, / , 9F12.0 )
0018            RETURN
0019            END

FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0002

                                  COMMON BLOCK / LODA    / MAP SIZE  000024
SYMBOL   LOCATION    SYMBOL   LOCATION   SYMBOL   LOCATION   SYMBOL   LOCATION   SYMBOL   LOCATION
XLOAD    000000      YLOAD    000004     XYLOD    000008     XMOM     00000C     YMOM     000010
XYMOM    000014      TORQ     000018     XQSHR    00001C     YQSHR    000020

                                  SCALAR MAP
SYMBOL   LOCATION    SYMBOL   LOCATION   SYMBOL   LOCATION   SYMBOL   LOCATION   SYMBOL   LOCATION
ICNT     0000DC      NSTP     0000E0     IOPT     0000E4     N        0000E8     FACT     0000EC

                                  SUBPROGRAMS CALLED
SYMBOL   LOCATION    SYMBOL   LOCATION   SYMBOL   LOCATION   SYMBOL   LOCATION   SYMBOL   LOCATION
IBCOM#   0000F0

                                  LABEL MAP
LABEL    LOCATION    LABEL    LOCATION   LABEL    LOCATION   LABEL    LOCATION   LABEL    LOCATION
  6080   0002E4

 TOTAL MEMORY REQUIREMENTS 000380 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//NODGEN EXEC FORTRAN(BCD,MAP)
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0001

          C*********************************** OVERLAY AA ********************************
0001            SUBROUTINE NODGEN (IWRT,X,Y,Z,SLOPA)
0002            COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,VEL(20)
0003            COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
               1  ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
0004            DIMENSION X(300),Y(300),Z(300),SLOPA(300)
          C
          C     GENERATE FLAT PLATE NODAL COORDINATES
0005            XC(11) = XC(11) + .00001
0006            NUMNP = 2*NX*NY
0007            NYA = 2*NX
0008            Y1LGT = .28867 + Y1OFF
0009            Y2LGT = .86657 + Y2OFF
0010            DO 300 I = 1,NUMNP
0011            IROW = (I-1)/NX + 1
0012            ICOL = I - I/NYA*NYA
```

```
      IF (ICOL) 80,70,80
   70 ICOL = NX
      IF (FLOAT(I/NX)/2.-FLOAT(I/(NX*2)) .LT. .01) ICOL = 2*NX
   80 CONTINUE
      IF (ICOL - NX/2) 100,100,85
   85 IF (ICOL- NX  ) 150,150,90
   90 IF ( ICOL - 3*NX/2) 200,200,95
   95 Y(I) = SIDE*( Y2LGT + Y1LGT+2.*Y2LGT *(ICOL-3*NX/2-1))
      Z(I) = HT
      GO TO 250
  100 Y(I) = Y2LGT *SIDE*(2*ICOL-1)
      Z(I) = 0.0
      GO TO 250
  150 Y(I) = SIDE*(Y1LGT + Y2LGT*(ICOL-1-NX/2)*2.)
      Z(I) = HT
      GO TO 250
  200 Y(I) =  Y2LGT*SIDE*(ICOL-1-NX)*2.
      Z(I) = 0.0
  250 X(I) = SIDE/2.*(IROW-1) + Y(I)*SIN(THETA/57.2958)
  300 CONTINUE
      IF ( IOPT .NE. 1 .AND. ITYP .EQ. 1) WRITE (6,6140) Y(NUMNP)
 6140 FORMAT (/// , 5X, 18HFLAT PANEL WIDTH # , F15.5 )
      IF ( IOPT .NE. 1) WRITE (6,6150) X(NUMNP-NX+1)
 6150 FORMAT ( /// ,14X, 8HLENGTH # , F15.5 )
      IF (ITYP - 2) 800,350,550
C
C     GENERATE CYLINDER COORDINATES
  350 CIRC =  Y2LGT*SIDE*NX
      RAD =  Y2LGT* SIDE*NX/6.28318
      IF ( IOPT .NE. 1) WRITE (6,6200) RAD
 6200  FORMAT ( /// ,10X, 17HCYLINDER RADIUS # , F15.5 )
 6000 FORMAT ( 6F15.5 )
      DO 400 I = 1,NUMNP
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0002
      THETA = (Y(I)/CIRC)*6.28318
      ABC = Z(I)
      Y(I) = (RAD-ABC)*COS(THETA)
  400 Z(I) = (RAD-ABC)*SIN(THETA)
  450 CONTINUE
      GO TO 800
C
C     GENERATE AIRFOIL COORDINATES
  550 CONTINUE
      CHORD =  Y2LGT*SIDE*NX/2.
      IF (IOPT .NE. 1) WRITE (6,6250) CHORD
 6250  FORMAT ( /// ,10X, 15HAIRFOIL CHORD #   , F15.5 )
C
      DO 640 I = 1,NUMNP
      FACT = 1.0
      IF (Y(I) - CHORD) 610,610,605
  605 Y(I) = 2.*CHORD - Y(I)
      FACT = -1.0
  610 XOC = Y(I)/CHORD
      CALL CURV (XOC,TOC,XC,TC,SLOP)
      SLOPA(I) = SLOP*FACT
      ZOLD = Z(I)
      Z(I) = TOC*CHORD*FACT
      IF (ZOLD - .001) 640,620,620
  620 Z(I) = Z(I) - HT*FACT
```

```
      IF (FACT) 624,622,622
  622 IF (Z(I)) 625,640,640
  624 IF (Z(I)) 640,640,625
  625 Z(I) = 0.0
      IF (Y(I) - CHORD/2.) 630,630,635
  630 Y(I) = ABS(HT/SLOPA(I))
      GO TO 640
  635 Y(I) = CHORD - ABS(HT/SLOPA(I))
  640 CONTINUE
  800 CONTINUE
      IF (IWRT .EQ. 1) RETURN
  500 CONTINUE
      WRITE (3) (I,X(I),Y(I),Z(I),I=1,NUMNP)                    TAPE 3
      RETURN
      END
```

FORTRAN IV MODEL 44 PS VERSION 3, LEVEL 4 DATE 71321 PAGE 0003

COMMON BLOCK / COMN / MAP SIZE 00006C

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	000000	NUMEL	000004	NLOD	000008	NLIN	00000C	NBAND	000010
NUMBLK	000014	NBN	000018	NEL	00001C				

COMMON BLOCK / ONE / MAP SIZE 0000E8

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	000000	HT	000004	THX	000008	THETA	00001C	Y1OFF	000020
Y2OFF	000024	THT	000028	DUM	00002C	TC	000030	XC	00005C
HED	000088	NX	000088	NY	00008C	IDP7	0000C0	ITYP	0000C4
IFACE	0000C8	IHOLE	0000CC	ITET	0000D0	ICNT	0000D4	NNEW	0000D8

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NYA	000168	Y1LGT	00016C	Y2LGT	000170	I	000174	IROW	000178
ICOL	00017C	CIRC	000180	RAD	000184	ABC	000188	CHORD	00018C
FACT	000190	XOC	000194	TOC	000198	SLOP	00019C	ZOLD	0001A0
IWRT	0001A4								

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
X	0001A8	Y	0001AC	Z	0001B0	SLOPA	0001B4		

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	0001B8	CURV	0001BC	SIN	0001C0	COS	0001C4		

LABEL MAP

LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
70	000328	80	0003A4	85	0003C6	90	0003DA	95	000402
100	00047C	150	0004D8	200	000546	250	0005A0	300	00060C
6140	00068C	6150	000704	350	00073C	6200	0007C8	6000	0007EE
400	000862	450	000896	550	00089C	6250	000900	605	00095A
610	000986	620	0009F0	622	000A26	624	000A4C	625	000A6C
630	000AA2	635	000AD2	640	000B02	800	000B1C	500	000B36

TOTAL MEMORY REQUIREMENTS 000C1C BYTES

COMPILER HIGHEST SEVERITY CODE WAS 0
//PLATGN EXEC FORTRAN(BCO,MAP)

```
          C********************************** OVERLAY AB *********************************
0001            SUBROUTINE PLATGN
0002            COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
0003            COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
               1  ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
0004            COMMON / MATL / E(4,5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
0005            COMMON / PLAT/  NA(1100),NB(1100),NC(1100),NST(2),INT(6),ND(1100)
0006            COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLD,NR1,NR2
0007            IF (ICNT .GT. 0) GO TO 1000
          C
          C     GENERATE VERTICAL PLATES
0008            NAODD = NX/2 + 1
0009            NBODD = 5*NX/2 + 1
0010            NCODD = NX + 1
0011            NAEVN = 1
0012            NBEVN = 2*NX + 1
0013            NCEVN = 3*NX/2 + 1
0014            NP = 0
0015            NK = 0
0016            DO 300 NV = 1,NX
0017            NP = NP + 1
0018            NK = NK + 1
0019            IF (NK - 1) 90,100,90
0020         90 NA(NP) = NAEVN + NV/2 - 1
0021            NB(NP) = NBEVN + NV/2 - 1
0022            NC(NP) = NCEVN + NV/2 - 1
0023            NK = 0
0024            GO TO 150
0025        100 NA(NP) = NAODD + NV/2
0026            NB(NP) = NBODD + NV/2
0027            NC(NP) = NCODD + NV/2
0028        150 CONTINUE
0029            NPIR = 2*(NY-1)
0030            DO 300 I = 2,NPIR
0031            NP = NP + 1
0032            NA(NP) = NC(NP-1)
0033            NC(NP) = NB(NP-1)
0034        300 NB(NP) = NA(NP) + 2*NX
          C
          C     GENERATE SKEW A PLATES
0035            NNEW(1) = NP + 1
0036            NAA = 1
0037            NBA = NX + 1
0038            NCA = NX/2 + 1
0039            NAB = 2*NX
0040            NBB = 3*NX
0041            NCB = 5*NX/2
0042            NP = NP + 1
0043            NA(NP) = NAA
0044            NB(NP) = NBA
0045            NC(NP) = NCA
0046            NOSA = NX/2 + NY - 1
0047            NPIR = 5
0048            DO 500 I = 2,NOSA
0049            NP = NP + 1
0050            IF (I - NX/2) 310,310,350
0051        310 NA(NP) = NAA + I - 1
0052            NB(NP) = NBA + I - 1
0053            NC(NP) = NCA + I - 1
0054            NK = 3
```

```
      GO TO 400
  350 NA(NP) = NAB + 2*NX*(I-NX/2-1)
      NB(NP) = NBB + 2*NX*(I-NX/2-1)
      NC(NP) = NCB + 2*NX*(I-NX/2-1)
      NK = 2
 6000 FORMAT (  5I5 )
  400 DO 450 J = 2,NPIR
      NP = NP + 1
      NA(NP) = NC(NP-1)
      NC(NP) = NB(NP-1)
      NK = NK + 1
      NB(NP) = NC(NP) + NX/2
      IF (NK .EQ. 4) NB(NP) = NB(NP) - 1
  450 IF (NK .EQ. 4) NK = 0
      IF (I .LT. NX/2) NPIR = NPIR + 4
      IF ((I .GE. NX/2 .AND. I .GE. NY) .OR. (I .GE. NY)) NPIR = NPIR-4
      IF (I. EQ. NY   ) NPIR = NPIR + 1
  500 IF (I. EQ. NX/2  ) NPIR = NPIR + 1
C
C     ZIP UP SEAM OF CYLINDER
      IF (ITYP - 2) 540,505,505
  505 DO 530 I = 1,NY
      NP = NP + 1
      NA(NP) = NCA + (I-1)*NX*2
      NC(NP) = NBA + (I-1)*NX*2
      NB(NP) = NC(NP) + NX - 1
      IF (I - NY) 510,530,510
  510 NP = NP + 1
      NA(NP) = NC(NP-1)
      NC(NP) = NB(NP-1)
      NB(NP) = NC(NP) + NX/2
  530 CONTINUE
C
C     GENERATE SKEW B PLATES
  540 NNEW(2) = NP + 1
      NAA = NX
      NBA = 2*NX
      NCA = NX/2
FORTRAN IV    MODEL 44  PS     VERSION 3,  LEVEL 4   DATE  71321                 PAGE  0003
      NAB = NAA + 1
      NBB = NBA + 1
      NCB = 5*NX/2 + 1
      NP = NP + 1
      NA(NP) = NAA
      NB(NP) = NBA
      NC(NP) = NCA
      NOSB = NX/2 + NY - 1
      NPIR = 5
      DO 700 I = 2,NOSB
      NP = NP + 1
      IF (I- NX/2) 545,545,550
  545 NA(NP) = NAA - I + 1
      NB(NP) = NBA - I + 1
      NC(NP) = NCA - I + 1
      NK = 1
      GO TO 600
  550 NA(NP) = NAB + 2*NX*(I-NX/2-1)
      NB(NP) = NBB + 2*NX*(I-NX/2-1)
      NC(NP) = NCB + 2*NX*(I-NX/2-1)
      NK = 0
```

```
  600 DO 650 J = 2,NPIR
      NP = NP + 1
      NA(NP) = NC(NP-1)
      NC(NP) = NB(NP-1)
      NK = NK + 1
      GO TO (610,620,610,630,605),NK
  605 NK = 1
  610 NB(NP) = NC(NP) + 3*NX/2
      GO TO 650
  620 NB(NP) = NC(NP) - NX/2 + 1
      GO TO 650
  630 NB(NP) = NC(NP) - NX/2
  650 CONTINUE
      IF (I .LT. NX/2) NPIR = NPIR + 4
      IF ((I .GE. NX/2 .AND. I .GE. NY) .OR. (I .GE. NY)) NPIR = NPIR-4
      IF (I. EQ. NY  ) NPIR = NPIR + 1
  700 IF (I. EQ. NX/2  ) NPIR = NPIR + 1
C
C     ZIP UP SEAM OF CYLINDER
      IF (ITYP - 2) 740,710,710
  710 DO 730 I = 1,NY
      NP = NP + 1
      NA(NP) = NCA + (I-1)*NX*2
      NC(NP) = NBA + (I-1)*NX*2
      NB(NP) = NC(NP) - NX + 1
      IF (I - NY) 720,730,720
  720 NP = NP + 1
      NA(NP) = NC(NP-1)
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0004

```
      NC(NP) = NB(NP-1)
      NB(NP) = NC(NP) + 3*NX/2
  730 CONTINUE
C
C     GENERATE LOWER FACE SHEET
  740 NNEW(3) = 0.0
      NNEW(4) = 0.0
      IF (IFACE)750,850,750
  750 NNEW(3) = NP + 1
      NAODD = 1
      NBODD = 1 + 2*NX
      NCODD = 1 + NX
      NK = 0
      NY1 = NX - 1
      K = 0
      DO 770 I = 1,NY1
      K = K + 1
      IF (K .EQ. 3) K = 1
      NP = NP + 1
      NA(NP) = NAODD + (I-1)/2
      NB(NP) = NBODD + (I-1)/2
      NC(NP) = NCODD + (I-1)/2
      IF (K .EQ. 2) NC(NP) = NC(NP) + 1
      NNPT = 2*(NY-1)
      DO 770 J = 2,NNPT
      NP = NP + 1
      NA(NP) = NC(NP-1)
      NC(NP) = NB(NP-1)
  770 NB(NP) = NA(NP) + 2*NX
C
C     ZIP UP SIDE OF CYLINDER
```

```
      IF (ITYP - 2) 800,780,780
  780 NP = NP + 1
      NA(NP) = NX/2
      NB(NP) = 5*NX/2
      NC(NP) = NX + 1
      DO 790 I = 2,NX
      NP = NP + 1
      NA(NP) = NC(NP-1)
      NC(NP) = NB(NP-1)
  790 NB(NP) = NA(NP) + 2*NX
C
C     GENERATE UPPER FACE SHEET
  800 NNEW(4) = NP + 1
      NAODD = NX/2 + 1
      NBODD = 5*NX/2 + 1
      NCODD = 3*NX/2 + 1
      NK = 1
      K = 0
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                PAGE 0005

      DO 810 I = 1,NY1
      K = K + 1
      IF (K .EQ. 3) K = 1
      NP = NP + 1
      NA(NP) = NAODD + (I-1)/2
      NB(NP) = NBODD + (I-1)/2
      NC(NP) = NCODD + (I-1)/2
      IF (K-2) 805,802,802
  802 NA(NP) = NA(NP) + 1
      NB(NP) = NB(NP) + 1
  805 NMPT = 2*(NY-1)
      DO 810 J = 2,NMPT
      NP = NP + 1
      NA(NP) = NC(NP-1)
      NC(NP) = NB(NP-1)
  810 NB(NP) = NA(NP) + 2*NX
C
C     ZIP UP SIDE OF CYLINDER
      IF (ITYP - 2) 850,820,820
  820 NP = NP + 1
      NA(NP) = NX/2 + 1
      NB(NP) = 5*NX/2    + 1
      NC(NP) = 2*NX
      DO 830 I = 2,NX
      NP = NP + 1
      NA(NP) = NC(NP-1)
      NC(NP) = NB(NP-1)
  830 NB(NP) = NA(NP) + 2*NX
  850 NUMEL = NP
      WRITE (6,6200) NAEVN,NNEW
 6200 FORMAT ( /// , 5X, 22HFIRST VERTICAL PLATE # ,I4, 5X,8HSKEW A #,
     1 I4, 5X, 8HSKEW B #, I4, 5X,10HLWR FACE #,I4, 5X,12HUPPER FACE #,
     2 I4 )
      J = 1
      NODH = NUMNP/2 - NX + 1 + NX/4
      IF (IHOLE .GT. 1) NODH = IHOLE
      DO 900 I = 1,NUMEL
      MD(I)=0
      IF (I .EQ. NNEW(J)) J = J + 1
      XNU = 1. - E(4, J)*E(2,J)/E(1,J) *E(4,J)
      E11 = E(1,J)/XNU
```

```
      E22 = E(2,J)/XNU
      E12 = E(2,J)*E(4,J)/XNU
      E33 = E(3,J)
      IF (IHOLE-1) 890,860,860
  860 IF (NA(I )-NODH) 865,885,865
  865 IF (NB(I )-NODH) 870,885,870
  870 IF (NC(I )-NODH) 890,885,890
  885 E11 = 0.0
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0006

```
      E22 = 0.0
      E12 = 0.0
      E33 = 0.0
      WRITE (6,6150) NODH
 6150 FORMAT ( ///, 5X, 25HPLATES CONNECTING TO NODE, I5, 10H HAVE BEEN
     1   40H GIVEN ZERO STIFFNESS TO SIMULATE A HOLE   )
      IHOLE = NODH
  890 THICK = THK(J)
  900 WRITE (3) I,NA(I),NB(I),NC(I),ND(I),E22,E11,E12,E33,THICK,J        TAPE 3
      GO TO 1020
 1000 CONTINUE
      K = 1
      DO 1010 J = 1,NUMEL
      NR2=J
      READ(2'NR2)I,NA(I),NB(I),NC(I),ND(I),E22,E11,E12,E33,THICK,IMTL    TAPE 2
      IF (J .EQ. NNEW(K)) K = K + 1
      THICK = THK(K)
 1010 WRITE (3) I,NA(I),NB(I),NC(I),ND(I),E22,E11,E12,E33,THICK,IMTL     TAPE 3
      WRITE (6,6100) (THK(I),I=1,IMTL)
 6100 FORMAT ( 6F15.5)
 1020 CONTINUE
      RETURN
      END
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0007

COMMON BLOCK / COMN / MAP SIZE 00006C

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	000000	NUMEL	000004	NLOD	000008	NLIN	00000C	MBAND	000010
NUMBLK	000014	NBN	000018	NEL	00001C				

COMMON BLOCK / ONE / MAP SIZE 0000E8

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIZE	000000	HT	000004	THK	000008	THETA	00001C	Y1OFF	000020
Y2OFF	000024	THT	000028	DUM	00002C	TC	000030	XC	00005C
HEO	000088	NX	0000B8	NY	0000BC	IOPT	0000C0	ITYP	0000C4
IFACE	0000C8	IHOLE	0000CC	ITET	0000D0	ICNT	0000D4	NNEW	0000D8

COMMON BLOCK / MATL / MAP SIZE 0000DC

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
E	000000	FXT	000050	FYT	000064	FXC	000078	FYC	00008C
SSS	0000A0	CNS	0000B4						

COMMON BLOCK / PLAT / MAP SIZE 0044E0

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NA	000000	NB	001130	NC	002260	NST	003390	INT	003398
ND	0033B0								

```
                                        COMMON BLOCK / RAX     / MAP SIZE  000024
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
NR4       000000      NR4LOD    000004      NR4STF    000008      NR4DIA    00000C      NREC      000010
NRECST    000014      NRECLD    000018      NR1       00001C      NR2       000020

                                        SCALAR MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
NAODD     000218      NBODD     00021C      NCODD     000220      NAEVN     000224      NBEVN     000228
NCEVN     00022C      NP        000230      NK        000234      NV        000238      NPIR      00023C
I         000240      NAA       000244      NBA       000248      NCA       00024C      NAB       000250
NBB       000254      NCB       000258      NOSA      00025C      J         000260      NOSB      000264
NY1       000268      K         00026C      NMPT      000270      NODH      000274      XNU       000278
E11       00027C      E22       000280      E12       000284      E33       000288      THICK     00028C
INTL      000290

                                        SUBPROGRAMS CALLED
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
IBCOM#    000294

                                        LABEL MAP
LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION
   90     0003A0        100     000416        150     00046E        300     0004C8        310     000600
  350     00065E       6000     000706        400     000714        450     0007A6        500     000882
  505     0008D6        510     00096A        530     0009C6        540     0009E0        545     000AE4
  550     000B42        600     000BEA        605     000C78        610     000C84        620     000CC0

 FORTRAN IV    MODEL 44   PS         VERSION 3,  LEVEL 4    DATE  71321                    PAGE  0008

  630     000CFC        650     000D2E        700     000DF0        710     000E44        720     000ED8
  730     000F3A        740     000F54        750     000F78        770     0010E2        780     00112E
  790     0011D0        800     0011FA        802     001318        805     001348        810     0013A2
  820     0013EE        830     001498        850     0014C2       6200     0014FC        860     001672
  865     001696        870     0016BA        885     0016DE       6150     001720        890     00178E
  900     0017A2       1000     00184C       1010     001936       6100     001A24       1020     001A32

  TOTAL MEMORY REQUIREMENTS 001A88 BYTES

 COMPILER HIGHEST SEVERITY CODE WAS  0
 //LODGEN EXEC FORTRAN(BCD,MAP)

 FORTRAN IV    MODEL 44   PS         VERSION 3,  LEVEL 4    DATE  71321                    PAGE  0001

            C************************************* OVERLAY AC **********************************
                  SUBROUTINE LODGEN (X,Y,Z,SLOPA)
                  COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
                  COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
                 1 ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
                  COMMON / LODA / XLOAD,YLOAD,XYLOD,XMOM,YMOM,XYMOM,TORQ,XQSHR,YQSHR
                  DIMENSION X(300),Y(300),Z(300),SLOPA(300)
                  COMMON / LODS/  CODE(300),UX(300),UY(300),UZ(300)
                  NN3 = 3*NUMNP
                  DO 50 I = 1,NUMNP
                  UX(I) = 0.0
                  UY(I) = 0.0
               50 UZ(I) = 0.0
                  N1 = NX/2 + 1
                  N2 = NX + 1
                  N3 = NX/2
                  N4 = 2*NX
                  KONST = 2*NX*(NY-1)
                  NXND = 2*NX + (2*NX-1)*4
```

```
0018          NYND = 2*NY + (2*NY-1)*4
0019          Y1LGT = .28867 + Y1OFF
0020          Y2LGT = .86657 + Y2OFF
0021          XLGT = ((NX-1)*Y2LGT  +  Y1LGT)*SIDE
0022          YLGT = (FLOAT(NY)-.5)*SIDE
0023          IUNIT = 0
0024          IF (ABS(XLOAD) - .001) 60,102,102
0025       60 IF (ITYP - 2) 62,75,75
0026       62 IF (ABS(YLOAD) - .001)  65,110,110
0027       65 IF (ABS(XYLOD) - .001)  70,170,170
0028       70 IF (ABS(YMOM ) - .001)  75,270,270
0029       75 IF (ABS(XMOM ) - .001)  80,320,320
0030       80 IF (ITYP - 2) 82,85,85
0031       82 IF (ABS(XYMOM) - .001)  83,360,360
0032       83 IF (ABS(XQSHR) - .001) 84,500,500
0033       84 IF (ABS(YQSHR) - .001) 85,600,600
0034       85 IF (ABS(TORQ ) - .001)  800,650,650
        C
        C     FLAT PLATE END LOADS
        C     NX
0035      102 PX    = XLOAD*XLGT/NXND
0036          IF (ITYP - 2) 105,103,103
0037      103 PX = XLOAD/NXND
0038      105 DO 100 I = 1,N4
0039          UX(I) = UX(I) - PX
0040      100 UX(I+KONST) = UX(I+KONST) + PX
0041          IF (IUNIT) 60,60,800
        C
        C     NY
0042      110 PY    = YLOAD*YLGT/NYND
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4    DATE  71321                  PAGE  0002
0043          DO 150 I = 1,NY
0044          IAB = 2*NX*(I-1)
0045          UY(N1+IAB) = UY(N1+IAB) - PY
0046          UY(N2+IAB) = UY(N2+IAB) - PY
0047          UY(N3+IAB) = UY(N3+IAB) + PY
0048      150 UY(N4+IAB) = UY(N4+IAB) + PY
0049          IF (IUNIT) 65,65,800
        C
        C     NXY
0050      170 PXYA = XYLOD*XLGT/NXND
0051          PXYB = XYLOD*YLGT/NYND
0052          DO 200 I = 1,N4
0053          UY (I) = UY(I) - PXYA
0054      200 UY(I+KONST) = UY(I+KONST) + PXYA
0055          DO 250 I = 1,NY
0056          IAB = 2*NX*(I-1)
0057          UX(N1+IAB) = UX(N1+IAB) - PXYB
0058          UX(N2+IAB) = UX(N2+IAB) - PXYB
0059          UX(N3+IAB) = UX(N3+IAB) + PXYB
0060      250 UX(N4+IAB) = UX(N4+IAB) + PXYB
0061          IF (IUNIT) 70,70,800
        C
        C     MY
0062      270 PMY   = YMOM *YLGT/(HT*NY)
0063          DO 300 I = 1,NY
0064          IAB = 2*NX*(I-1)
0065          UY(N1+IAB) = UY(N1+IAB) + PMY
0066          UY(N2+IAB) = UY(N2+IAB) - PMY
0067          UY(N3+IAB) = UY(N3+IAB) + PMY
0068      300 UY(N4+IAB) = UY(N4+IAB) - PMY
```

```
      IF (IUNIT) 75,75,800
C
C     MX
  320 IF (ITYP - 2) 325,335,335
  325 PMX  = XMOM *XLGT/(HT*NX)
      DO 330 I = 1,N3
      UX(I        ) = UX(I        ) - PMX
      UX(I+  NX/2) = UX(I+  NX/2) + PMX
      UX(I+  NX  ) = UX(I+  NX  ) - PMX
      UX(I+3*NX/2) = UX(I+3*NX/2) + PMX
      UX(I+KONST        ) = UX(I+KONST        ) + PMX
      UX(I+KONST+  NX/2) = UX(I+KONST+  NX/2) - PMX
      UX(I+KONST+  NX  ) = UX(I+KONST+  NX  ) + PMX
  330 UX(I+KONST+3*NX/2) = UX(I+KONST+3*NX/2) - PMX
      GO TO 350
C     BENDING MOMENT ON CYLINDER AND AIRFOIL
  335 SUM = 0.0
      DO 340 I = 1,N4
  340 SUM = SUM + Z(I)**2*3.
```



```
      PMX = XMOM/SUM/3.
      DO 345 I = 1,N4
      UX(I) = UX(I) + PMX*Z(I)
  345 UX(I+KONST) = UX(I+KONST) - PMX*Z(I)
  350 IF (IUNIT) 80,80,800
C     MXY
  360 PMXYA= XYMOM*YLGT/(HT*NY)
      PMXYB= XYMOM*XLGT/(HT*NX)
C
C     LOAD ON X FACES
      DO 400 I = 1,N3
      UY(I        ) = UY(I        ) - PMXYA
      UY(I+  NX/2) = UY(I+  NX/2) + PMXYA
      UY(I+  NX  ) = UY(I+  NX  ) - PMXYA
      UY(I+3*NX/2) = UY(I+3*NX/2)+ PMXYA
      UY(I+KONST        ) = UY(I+KONST        ) + PMXYA
      UY(I+KONST+  NX/2) = UY(I+KONST+  NX/2) - PMXYA
      UY(I+KONST+  NX  ) = UY(I+KONST+  NX  ) + PMXYA
  400 UY(I+KONST+3*NX/2) = UY(I+KONST+3*NX/2) - PMXYA
C
C     LOAD ON Y FACES
      DO 450 I = 1,NY
      IAB = 2*NX*(I-1)
      UX(N1+IAB) = UX(N1+IAB) - PMXYB
      UX(N2+IAB) = UX(N2+IAB) + PMXYB
      UX(N3+IAB) = UX(N3+IAB) - PMXYB
  450 UX(N4+IAB) = UX(N4+IAB) + PMXYB
      IF (IUNIT) 83,83,800
C     QX
  500 PZ = XQSHR*XLGT/NXNO
      PPY = XQSHR*SIDE*(NY-1)*XLGT/(2.*HT*NX)
      DO 520 I = 1,N3
      UX(I        ) = UX(I        ) + PMY
      UZ(I        ) = UZ(I        ) + PZ
      UX(I+  NX/2) = UX(I+  NX/2) - PMY
      UZ(I+  NX/2) = UZ(I+  NX/2) + PZ
      UX(I+  NX  ) = UX(I+  NX  ) + PMY
      UZ(I+  NX  ) = UZ(I+  NX  ) + PZ
      UX(I+3*NX/2) = UX(I+3*NX/2) - PMY
```

```
      UZ(I+3*NX/2) = UZ(I+3*NX/2) + PZ
      UX(I+KONST        ) = UX(I+KONST        ) + PMY
      UZ(I+KONST        ) = UZ(I+KONST        ) - PZ
      UX(I+KONST+  NX/2) = UX(I+KONST+  NX/2) - PMY
      UZ(I+KONST+  NX/2) = UZ(I+KONST+  NX/2) - PZ
      UX(I+KONST+  NX  ) = UX(I+KONST+  NX  ) + PMY
      UZ(I+KONST+  NX  ) = UZ(I+KONST+  NX  ) - PZ
      UX(I+KONST+3*NX/2) = UX(I+KONST+3*NX/2) - PMY
  520 UZ(I+KONST+3*NX/2) = UZ(I+KONST+3*NX/2) - PZ
      IF (IUNIT) 84,84,800
```



```
C     QY
  600 PZ = YQSHR*YLGT/NYND
      PMX = YQSHR*SIDE*(.86657+Y2OFF)*YLGT/(2.*HT*NY)
      DO 620 I = 1,NY
      IAB = 2*NX*(I-1)
      UY(N1+IAB) = UY(N1+IAB) - PMX
      UZ(N1+IAB) = UZ(N1+IAB) + PZ
      UY(N2+IAB) = UY(N2+IAB) + PMX
      UZ(N2+IAB) = UZ(N2+IAB) + PZ
      UY(N3+IAB) = UY(N3+IAB) + PMX
      UZ(N3+IAB) = UZ(N3+IAB) - PZ
      UY(N4+IAB) = UY(N4+IAB) - PMX
  620 UZ(N4+IAB) = UZ(N4+IAB) - PZ
      IF (IUNIT) 85,85,800
C
C     GENERATE TORSION LOADS ON CYLINDER
  650 IF (ITYP - 2) 800,660,720
  660 RAD1 = SQRT(Y(1)**2+Z(1)**2)
      K = NX/2 + 1
      RAD2 = SQRT(Y(K)**2+Z(K)**2)
      PA = TORQ/(NX*(RAD1+RAD2**2/RAD1)*3.)
      PB = RAD2/RAD1*PA
      DO 700 I = 1,N4
      THETA = ATAN2(Z(I),Y(I))
      SINTH = SIN(THETA+1.57079 )
      COSTH = COS(THETA+1.57079)
      PX = PA
      IF (I .GT. NX/2 .AND. I .LE. NX) PX = PB
      IF (I .GT. 3*NX/2) PX = PB
      UY(I) = UY(I) + PX*COSTH
      UZ(I) = UZ(I) + PX*SINTH
      UY(I+KONST) = UY(I+KONST) - PX*COSTH
  700 UZ(I+KONST) = UZ(I+KONST) - PX*SINTH
      GO TO 800
C
C     GENERATE TORSION LOADS ON AIRFOIL
  720 NEND = 5*NX/4 + 1
      ARM = 0.0
      DO 750 I = 1,N4
      ANG = SLOPA(I)
  750 ARM = ARM - Y(I)*SIN(ANG) + Z(I)*COS(ANG)
      PA = TORQ/ARM/3.
      DO 760 I = 1,N4
      ANG = SLOPA(I)
      IF (Z(I) .LT. 0.0) ANG = ANG + 3.14159
      UY(I) = UY(I) + PA*COS(ANG)
      UZ(I) = UZ(I) + PA*SIN(ANG)
      UY(I+KONST) = UY(I+KONST) - PA*COS(ANG)
  760 UZ(I+KONST) = UZ(I+KONST) - PA*SIN(ANG)
```

```
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                          PAGE  0005

          800 CONTINUE
          850 WRITE (3) (UX(I),UY(I),UZ(I),I=1,NUMNP)                             TAPE 3
              DO 900 I = 1,NUMNP
          900 CODE(I) = 0.0
              IF (IHOLE .GT. 0) CODE(IHOLE) = 7.
              CODE(9 ) = 7.
              CODE(14) = 3.
              CODE(17) = 6.
              WRITE (3) (CODE(I),I=1,NUMNP)                                         TAPE 3
              RETURN
              END

FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                          PAGE  0006
```

COMMON BLOCK / CONN / MAP SIZE 00006C

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	000000	NUMEL	000004	NLOD	000008	NLIN	00000C	MBAND	000010
NUMBLK	000014	NBN	000018	NEL	00001C				

COMMON BLOCK / ONE / MAP SIZE 0000E8

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	000000	HT	000004	THK	000008	THETA	00001C	Y1OFF	000020
Y2OFF	000024	THT	000028	DUM	00002C	TC	000030	XC	00005C
HED	000088	NX	0000B8	NY	0000BC	IOPT	0000C0	ITYP	0000C4
IFACE	0000C8	IHOLE	0000CC	ITET	0000D0	ICNT	0000D4	NNEW	0000D8

COMMON BLOCK / LODA / MAP SIZE 000024

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
XLOAD	000000	YLOAD	000004	XYLOD	000008	XMOM	00000C	YMOM	000010
XYMOM	000014	TORQ	000018	XQSHR	00001C	YQSHR	000020		

COMMON BLOCK / LODS / MAP SIZE 0012C0

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CODE	000000	UX	000480	UY	000960	UZ	000E10		

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NN3	0001A8	I	0001AC	N1	0001B0	N2	0001B4	N3	0001B8
N4	0001BC	KONST	0001C0	NXND	0001C4	NYND	0001C8	Y1LGT	0001CC
Y2LGT	0001D0	XLGT	0001D4	YLGT	0001D8	IUNIT	0001DC	PX	0001E0
PY	0001E4	IAB	0001E8	PXYA	0001EC	PXYB	0001F0	PMY	0001F4
PMX	0001F8	SUM	0001FC	PMXYA	000200	PMXYB	000204	PZ	000208
RAD1	00020C	K	000210	RAD2	000214	PA	000218	PB	00021C
SINTH	000220	COSTH	000224	NEND	000228	ARM	00022C	ANG	000230

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
X	000234	Y	000238	Z	00023C	SLOPA	000240		

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	000244	SQRT	000248	ATAN2	00024C	SIN	000250	COS	000254

```
                                                LABEL MAP
LABEL   LOCATION    LABEL   LOCATION    LABEL   LOCATION    LABEL   LOCATION    LABEL   LOCATION
   50   00036A         60   000484         62   0004CC         65   0004E6         70   000500
   75   00051A         80   000534         82   00054C         83   000566         84   000580
   85   00059A        102   0005BA        103   000600        105   00062E        100   000652
  110   0006AA        150   00079E        170   0007F2        200   00086E        250   000976
  270   0009CA        300   000AC6        320   00081A        325   000832        330   000D04
  335   000074        340   000D88        345   000E04        350   000E4C        360   000E5E

FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4    DATE  71321                      PAGE  0007

  400   00105E        450   00118A        500   0011DE        520   00150A        600   001656
  620   001850        650   0018A4        660   0018C2        700   001AEA        720   001838
  750   001886        760   001C9E        800   001CEC        850   001CEC        900   001D54

 TOTAL MEMORY REQUIREMENTS 001E90 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//NODGAN EXEC FORTRAN(BCD,MAP)


FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4    DATE  71321                      PAGE  0001

             C************************************ OVERLAY AD ***********************************
0001               SUBROUTINE NODGAN (IWRT,X,Y,Z,SLOPA)
0002               COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
0003               COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
                 1  ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
0004               DIMENSION X(300),Y(300),Z(300),SLOPA(300)
             C
             C     GENERATE FLAT PLATE NODAL COORDINATES
0005               XC(11) = XC(11) + .00001
0006               NODLG = 4*NY - 1
0007               SINTH = SIN(THETA/57.2958)
0008               TANTH = TAN(THETA/57.2958)
0009               XOFF = (.28867 + Y1OFF)*SIDE*TANTH
0010               NUMNP = NX*NY + NX*(3*NY-1)
0011               X1 = (THT - HT)/THT*SIDE
0012               X2 = (SIDE/2. + XOFF)*HT/THT
0013               Y1 = (.86603+Y2OFF)*SIDE - (.57734+Y1OFF)*SIDE*HT/THT
0014               Y2 = (.28867 + Y1OFF)*SIDE*HT/THT
0015               YINC = SIDE*(.86603+Y2OFF)
0016               Y2LGT = .86603 + Y2OFF
0017               IK = 0
0018               DO 300 I = 1,NX
0019               IF (IK) 110,110,120
0020           110 I1 = (I-1)*NODLG + 1
0021               Y(I1) = Y1 + YINC*(I-1)
0022               Y(I1+1) = Y2 + YINC*(I-1)
0023               Y(I1+2) = YINC*(I-1)
0024               Y(I1+3) = Y(I1+1)
0025               X(I1) = Y(I1)*SINTH
0026               X(I1+1) = X1/2. + Y(I1+1)*SINTH
0027               X(I1+2) = SIDE/2. + Y(I1+2) *SINTH
0028               X(I1+3) = SIDE - X1/2. + Y(I1+3)*SINTH
0029               Z(I1) = HT
0030               Z(I1+1) = HT
0031               Z(I1+2) = 0.0
0032               Z(I1+3) = HT
0033               IK = 1
0034               I1 = I1 + 3
0035               GO TO 150
```

```
0036          120 I1 = (I-1)*NODLG + 1
0037              Y(I1) = YINC*(I-1)
0038              Y(I1+1) = Y(I1) + Y2
0039              Y(I1+2) = Y(I1) + Y1
0040              Y(I1+3) = Y(I1+1)
0041              X(I1) = Y(I1)*SINTH
0042              X(I1+1) = X2 + Y(I1+1)*SINTH
0043              X(I1+2) = SIDE/2. + Y(I1+2)*SINTH
0044              X(I1+3) = X(I1+1) + X1 + Y(I1+3)*SINTH
0045              Z(I1) = 0.0
```



```
0046              Z(I1+1) = HT
0047              Z(I1+2) = HT
0048              Z(I1+3) = HT
0049              IK = 0
0050              I1 = I1 + 3
0051          150 DO 300 J = 5,NODLG
0052              I1 = I1 + 1
0053              X(I1) = X(I1-4) + SIDE
0054              Y(I1) = Y(I1-4)
0055          300 Z(I1) = Z(I1-4)
0056              IF (IOPT .NE. 1) WRITE (6,6150) X(NODLG)
0057         6150 FORMAT ( /// ,14X, 8HLENGTH # , F15.5 )
0058              IF (ITYP - 2) 800,350,550
             C
             C    GENERATE CYLINDER COORDINATES
0059          350 CIRC = Y2LGT*SIDE*NX
0060              RAD =  Y2LGT*SIDE*NX/6.28318
0061              IF (IOPT .NE. 1) WRITE (6,6200) RAD
0062         6200  FORMAT ( /// ,10X, 17HCYLINDER RADIUS # , F15.5 )
0063         6000 FORMAT ( 6F15.5 )
0064              DO 400 I = 1,NUMNP
0065              THETA = (Y(I)/CIRC)*6.28318
0066              ABC = HT - Z(I)
0067              Y(I) = (RAD-ABC)*COS(THETA)
0068          400 Z(I) = (RAD-ABC)*SIN(THETA)
0069          450 CONTINUE
0070              GO TO 800
             C
             C    GENERATE AIRFOIL COORDINATES
0071          550 CONTINUE
0072              CHORD = Y2LGT*SIDE*NX/2.
0073              IF (IOPT .NE. 1) WRITE (6,6250) CHORD
0074         6250  FORMAT ( /// ,10X, 15HAIRFOIL CHORD #   , F15.5 )
             C
0075              DO 640 I = 1,NUMNP
0076              FACT = 1.0
0077              IF (Y(I) - CHORD) 610,610,605
0078          605 Y(I) = 2.*CHORD - Y(I)
0079              FACT = -1.0
0080          610 XOC = Y(I)/CHORD
0081              CALL CURV (XOC,TOC,XC,TC,SLOP)
0082              SLOPA(I) = SLOP*FACT
0083              ZOLD = Z(I)
0084              Z(I) = TOC*CHORD*FACT
0085              IF (ZOLD - .001) 640,620,620
0086          620 Z(I) = Z(I) - HT*FACT
0087              IF (FACT) 624,622,622
0088          622 IF (Z(I)) 625,640,640
0089          624 IF (Z(I)) 640,640,625
```

```
FORTRAN IV     MODEL 44   PS        VERSION 3,  LEVEL 4   DATE  71321                          PAGE  0003

           625 Z(I) = 0.0
               IF (Y(I) - CHORD/2.) 630,630,635
           630 Y(I) = ABS(HT/SLOPA(I))
               GO TO 640
           635 Y(I) = CHORD - ABS(HT/SLOPA(I))
           640 CONTINUE
         C
           800 CONTINUE
          6020 FORMAT ( 60X, 10HNODAL DATA  )
               DO 500 I = 1,NUMNP
          6010 FORMAT (I4,20X,4F12.5 )
           500 CONTINUE
         C                                                                   TAPE 3
               IF (IWRT .EQ. 1) RETURN
               WRITE (3) (I,X(I),Y(I),Z(I),I=1,NUMNP)
               RETURN
               END

FORTRAN IV     MODEL 44   PS        VERSION 3,  LEVEL 4   DATE  71321                          PAGE  0004
                                       COMMON BLOCK / COMN    / MAP SIZE  00006C
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
NUMNP     000000        NUMEL     000004        NLOD      000008        NLIN      00000C        NBAND     000010
NUMBLK    000014        NBN       000018        NEL       00001C

                                       COMMON BLOCK / ONE     / MAP SIZE  0000E8
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
SIDE      000000        HT        000004        THK       000008        THETA     00001C        Y1DFF     000020
Y2DFF     000024        THT       000028        DUM       00002C        TC        000030        XC        00005C
HED       000088        NX        000088        NY        0000BC        IOPT      0000C0        ITYP      0000C4
IFACE     0000C8        IHOLE     0000CC        ITET      0000D0        ICNT      0000D4        NNEW      0000D8

                                       SCALAR MAP
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
NODLG     000154        SINTH     000158        TANTH     00015C        XOFF      000160        X1        000164
X2        000168        Y1        00016C        Y2        000170        YINC      000174        Y2LGT     000178
IK        00017C        I         000180        I1        000184        J         000188        CIRC      00018C
RAD       000190        ABC       000194        CHORD     000198        FACT      00019C        XOC       0001A0
TOC       0001A4        SLOP      0001A8        ZOLD      0001AC        IWRT      0001B0

                                       ARRAY MAP
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
X         0001B4        Y         0001B8        Z         0001BC        SLOPA     0001C0

                                       SUBPROGRAMS CALLED
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
IBCOM#    0001C4        CURV      0001C8        SIN       0001CC        TAN       0001D0        COS       0001D4

                                       LABEL MAP
LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION
  110     0003D0          120     00052E          150     00063C          300     000684         6150     0006FC
  350     000734         6200     0007C0         6000     0007E6          400     00085E          450     000892
  550     000898         6250     0008FC          605     000956          610     000982          620     0009EC
  622     000A22          624     000A48          625     000A68          630     000A9E          635     000ACE
  640     000AFE          800     000B18         6020     000B18         6010     000B3A          500     000B4C

 TOTAL MEMORY REQUIREMENTS 000C6C BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//PLATGA EXEC FORTRAN(BCD,MAP)
```

```
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0001

C************************************* OVERLAY AE *********************************
      SUBROUTINE PLATGA
      COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
      COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
     1  ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
      COMMON / MATL / E(4,5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
      COMMON / PLAT/  NA(1100),NB(1100),NC(1100),NST(2),INT(6),ND(1100)
      COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREL,NRECST,NRECLD,NR1,NR2
      IF (ICNT .GT. 0) GO TO 1000
      NAEVN = 1
C     GENERATE VERTICAL LEGS
      NST(1) = 2
      NST(2) = 4*NY
      NPIR = 2*NY - 2
      INT(1) =  2
      INT(2) = -1
      INT(3) =  4
      INT(4) = -3
      INT(5) =  2
      NLEG = 8*NY - 2
      K = 0
      NP = 0
      DO 300 I = 1,NX
      NP = NP + 1
      IF (K) 210,210,220
  210 NA(NP) = NST(1) + NLEG*(I-1)/2
      NK = 1
      K = 1
      GO TO 230
  220 NA(NP) = NST(2) + NLEG*(I-2)/2
      NK = 3
      K = 0
  230 NB(NP) = NA(NP) + INT(NK)
      NC(NP) = NB(NP) + INT(NK+1)
      ND(NP) = 0
      IF (NK .EQ. 3) ND(NP) = NC(NP) + INT(NK+2)
      DO 300 J = 2,NPIR
      NP = NP + 1
      NK = NK + 2
      IF (NK .EQ. 5) NK = 1
  250 NA(NP) = NC(NP-1)
      IF (NK .EQ. 1) NA(NP) = ND(NP-1)
  260 NB(NP) = NA(NP) + INT(NK)
      NC(NP) = NB(NP) + INT(NK+1)
      ND(NP) = 0
      IF (NK .EQ. 1) GO TO 300
      ND(NP) = NC(NP) + INT(NK+2)
  300 CONTINUE
C     GENERATE SKEW A LEGS
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0002

      NNEW(1) = NP + 1
      NLEG = 8*NY - 2
      NST(1) = 4*NY
      NST(2) = 4*NY + (NX/2-1)*NLEG + 3
      INT(1) = -4*NY + 2
      INT(2) =  4*NY - 1
      INT(3) = INT(1) + 1
      INT(4) = -2
      INT(5) =  1
```

```
      NP = NP + 1
      NA(NP) = NST(1)
      NB(NP) = NA(NP) + INT(3)
      NC(NP) = NB(NP) + INT(4)
      ND(NP) = NC(NP) + INT(5)
      NPIR = 5
      NOSA = NX/2 + NY - 1
      DO 500 I = 2,NOSA
      NP = NP + 1
      IF (I-NX/2) 350,350,360
  350 NA(NP) = NST(1) + (I-1)*NLEG
      NK = 3
      GO TO 370
  360 NA(NP) = NST(2) + (I-NX/2-1)*4
      NK = 1
  370 NB(NP) = NA(NP) + INT(NK)
      NC(NP) = NB(NP) + INT(NK+1)
      ND(NP) = 0
      IF (NK .EQ. 3) ND(NP) = NC(NP) + INT(NK+2)
      DO 450 J = 2,NPIR
      NP = NP + 1
      NK = NK + 2
      IF (NK .EQ. 5) NK = 1
  410 NA(NP) = NC(NP-1)
      IF (NK .EQ. 1) NA(NP) = ND(NP-1)
  420 NB(NP) = NA(NP) + INT(NK)
      NC(NP) = NB(NP) + INT(NK+1)
      ND(NP) = 0
      IF (NK .EQ. 1) GO TO 450
      ND(NP) = NC(NP) + INT(NK+2)
  450 CONTINUE
      IF (I .LT. NX/2) NPIR = NPIR + 4
      IF ((I .GE. NX/2 .AND. I .GE. NY) .OR. (I .GE. NY)) NPIR = NPIR-4
      IF (I .EQ. NY) NPIR = NPIR + 1
      IF (I .EQ. NX/2) NPIR = NPIR + 1
  500 CONTINUE
      IF (ITYP .EQ. 1) GO TO 540
C     ZIP UP SIDE OF CYLINDER
      NDA = 2
      NDB = (4*NY-1)*(NX-1) +3
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                PAGE 0003
      DO 530 I = 1,NY
      NP = NP + 1
      NA(NP) = NDA + (I-1)*4
      NB(NP) = NDB + (I-1)*4
      NC(NP) = NA(NP) + 1
      ND(NP) = 0
      IF (I .EQ. NY) GO TO 530
      NP = NP + 1
      NA(NP) = NC(NP-1)
      NB(NP) = NB(NP-1) + 2
      NC(NP) = NB(NP-1)
      ND(NP) = NC(NP) + 1
  530 CONTINUE
C     GENERATE SKEW B LEGS
  540 NNEW(2) = NP + 1
      NST(1) = NLEG*(NX/2-1) + 1
      NST(2) = 3
      INT(1) = 4*NY
      INT(2) = -1
      INT(3) = 4*NY + 1
```

```
      INT(4) = -4*NY
      INT(5) = 1
      NOSB = NX/2 + NY - 1
      NP = NP + 1
      NA(NP) = NST(1)
      NB(NP) = NA(NP) + INT(1)
      NC(NP) = NB(NP) + INT(2)
      ND(NP) = 0
      NPIR = 5
      DO 700 I = 2,NOSB
      NP = NP + 1
      IF (I-NX/2) 550,550,560
  550 NA(NP) = NST(1) - (I-1)*NLEG
      NK = 1
      GO TO 570
  560 NA(NP) = NST(2) + (I-NX/2-1)*4
      NK = 3
  570 NB(NP) = NA(NP) + INT(NK)
      NC(NP) = NB(NP) + INT(NK+1)
      ND(NP) = 0
      IF (NK .EQ. 3) ND(NP) = NC(NP) + INT(NK+2)
      DO 650 J = 2, NPIR
      NP = NP + 1
      NK = NK + 2
      IF (NK .EQ. 5) NK = 1
  610 NA(NP) = NC(NP-1)
      IF (NK .EQ. 1) NA(NP) = ND(NP-1)
  620 NB(NP) = NA(NP) + INT(NK)
      NC(NP) = NB(NP) + INT(NK+1)
```



```
      ND(NP) = 0
      IF (NK .EQ. 1) GO TO 650
      ND(NP) = NC(NP) + INT(NK+2)
  650 CONTINUE
      IF (I .LT. NX/2) NPIR = NPIR + 4
      IF ((I .GE. NX/2 .AND. I .GE. NY) .OR. (I .GE. NY)) NPIR = NPIR-4
      IF (I .EQ. NY) NPIR = NPIR + 1
      IF (I .EQ. NX/2) NPIR = NPIR + 1
  700 CONTINUE
      IF (ITYP .EQ. 1) GO TO 740
C     ZIP UP SIDE OF CYLINDER
      NOA = 3
      NOB = (4*NY-1)*(NX-1) + 1
      DO 730 I = 1,NY
      NP = NP + 1
      NA(NP) = NOB + (I-1)*4
      NB(NP) = NOA + (I-1)*4
      NC(NP) = NA(NP) + 1
      ND(NP) = NC(NP) + 1
      IF (I .EQ. NY) GO TO 730
      NP = NP + 1
      NA(NP) = ND(NP-1)
      NB(NP) = NB(NP-1) +1
      NC(NP) = NB(NP-1)
      ND(NP) = 0
  730 CONTINUE
  740 NUMEL = NP
      WRITE (6,6200) NAEVN,NNEW(1),NNEW(2)
```

```
0169          6200 FORMAT ( /// , 5X, 22HFIRST VERTICAL PLATE # ,I4, 5X,8HSKEW A #,
                  1  I4, 5X, 8HSKEW B #, I4  )
0170               J = 1
0171               NODH = (4*NY-1)*NX/2 + 2*NY - 1
0172               IF (IHOLE .GT. 1) NODH = IHOLE
0173               DO 900 I = 1,NUMEL
0174               IF (I .EQ. NNEW(J)) J = J + 1
0175               XNU = 1. - E(4, J)*E(2,J)/E(1,J) *E(4,J)
0176               E11 = E(1,J)/XNU
0177               E22 = E(2,J)/XNU
0178               E12 = E(1,J)*E(4,J)/XNU
0179               E33 = E(3,J)
0180               IF (IHOLE-1) 890,860,860
0181           860 IF (NA(I )-NODH) 865,885,865
0182           865 IF (NB(I )-NODH) 870,885,870
0183           870 IF (NC(I )-NODH) 880,885,880
0184           880 IF (ND(I)-NODH) 890,885,890
0185           885 E11 = 0.0
0186               E22 = 0.0
0187               E12 = 0.0
0188               E33 = 0.0
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                         PAGE  0005

0189               IHOLE = NODH
0190               WRITE (6,6150) NODH
0191          6150 FORMAT ( ///, 5X, 25HPLATES CONNECTING TO NODE, I5, 10H HAVE BEEN
                 1    40H GIVEN ZERO STIFFNESS TO SIMULATE A HOLE    )
0192           890 THICK = THK(J)
0193          6000 FORMAT ( 6I5 )
0194           900 WRITE (3) I,NA(I),NB(I),NC(I),ND(I),E22,E11,E12,E33,THICK,J        TAPE 3
0195               GO TO 1020
0196          1000 CONTINUE
0197               K = 1
0198               DO 1010 J = 1,NUMEL
0199               NR2=J
0200               READ(2'NR2)I,NA(I),NB(I),NC(I),ND(I),E22,E11,E12,E33,THICK,INTL     TAPE 2
0201               IF (J .EQ. NNEW(K)) K = K + 1
0202               THICK = THK(K)
0203          1010 WRITE (3) I,NA(I),NB(I),NC(I),ND(I),E22,E11,E12,E33,THICK,INTL     TAPE 3
0204               WRITE (6,6100) (THK(I),I=1,INTL)
0205          6100 FORMAT ( 6F15.5)
0206          1020 CONTINUE
0207               RETURN
0208               END
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                         PAGE  0006


                                     COMMON BLOCK / COMN    / MAP SIZE  00006C
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
NUMNP     000000        NUMEL     000004        NLOD      000008        NLIN      00000C        MBAND     000010
NUMBLK    000014        NBN       000018        NEL       00001C

                                     COMMON BLOCK / ONE     / MAP SIZE  0000E8
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
SIDE      000000        HT        000004        THK       000008        THETA     00001C        Y1OFF     00002C
Y2OFF     000024        THT       000028        DUM       00002C        TC        000030        XC        00005C
HED       000085        NX        000088        NY        00008C        IOPT      0000C0        ITYP      0000C4
IFACE     0000C8        IHOLE     0000CC        ITER      0000D0        ICNT      0000D4        NNEW      0000D8
```

```
                                        COMMON BLOCK / MATL   / MAP SIZE  0000DC
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
E         000000        FXT       000050        FYT       000064        FXC       000078        FYC       00008C
SSS       0000A0        CMS       000084

                                        COMMON BLOCK / PLAT   / MAP SIZE  0044E0
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
NA        000000        NB        001130        NC        002260        NST       003390        INT       003398
ND        0033B0

                                        COMMON BLOCK / RAX    / MAP SIZE  000024
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
NR4       000000        NR4LOD    000004        NR4STF    000008        NR4DIA    00000C        NREC      000010
NRECST    000014        NRECLD    000018        NR1       00001C        NR2       000020

                                        SCALAR MAP
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
NAEVN     0001FC        NPIR      000200        NLEG      000204        K         000208        NP        00020C
I         000210        NK        000214        J         000218        NOSA      00021C        NOA       000220
NOB       000224        NOSB      000228        NODH      00022C        XNU       000230        E11       000234
E22       000238        E12       00023C        E33       000240        THICK     000244        IMTL      000248

                                        SUBPROGRAMS CALLED
SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
IBCOM#    00024C

                                        LABEL MAP
LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION
  210     000344          220     000392          230     0003DA          250     000482          260     0004BC
  300     000526          350     000694          360     0006D0          370     000716          410     00078E
  420     0007F8          450     000862          500     00094E          530     000A86          540     000AA0
  550     0008C0          560     000BFE          570     000C44          610     000CEC          620     000D26
  650     000D90          700     000E7C          730     000F84          740     000FCE         6200     001010
  860     001154          865     001178          870     00119C          880     0011C0          885     0011DC
 6150     001228          890     00128A         6000     00129E          900     0012AC         1000     001354

  FORTRAN IV     MODEL 44   PS        VERSION 3,  LEVEL 4    DATE  71321                          PAGE  0007

   1010    00143E          6100    00152C           1020    00153A

   TOTAL MEMORY REQUIREMENTS 0015A0 BYTES

COMPILER HIGHEST SEVERITY CODE WAS   0
//LODGAN EXEC FORTRAN(BCD,MAP)

 FORTRAN IV     MODEL 44   PS        VERSION 3,  LEVEL 4    DATE  71321                          PAGE  0001

               C************************************** OVERLAY AF ***********************************
  0001                SUBROUTINE LODGAN (X,Y,Z,SLOPA)
  0002                COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
  0003                COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
                    1   ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
  0004                COMMON / LODA / XLOAD,YLOAD,XYLOD,XMOM,YMOM,XYMOM,TORQ,XQSHR,YQSHR
  0005                DIMENSION X(300),Y(300),Z(300),SLOPA(300)
  0006                COMMON / LODS/  CODE(300),UX(300),UY(300),UZ(300)
  0007                COMMON /DUMP / ITEST
  0008                NUMNP = NX*NY + NX*(3*NY-1)
  0009                DO 40  I = 1,NUMNP
  0010             40 CODE (I) = 0.0
  0011                DO 50 I = 1,NUMNP
```

```
      UX(I) = 0.0
      UY(I) = 0.0
   50 UZ(I) = 0.0
      N1 = 4*(NY-1)
      N2 = (4*NY-1)*(NX-1) - 1
      NA = 4*NY-2
      Y2 = (.28867+Y1OFF)*SIDE*HT/THT
      NXND = 2*NX + 5*(2*NX-1) - 1
      NYND = 2*NY + 5*(2*NY-1) - 1
      XLGT  = (NX-1)*SIDE*(.86657+Y2OFF) + Y2
      YLGT  = (FLOAT(NY) - .5)*SIDE
      N4 = 2*NY
      KONST = 2*NY*(NX-1)
      IUNIT = 0
      IF (ITYP .GT. 1) GO TO 37
      XFIX = 3.0
      YFIX = 3.0
      IF (ABS(XQSHR) .LT. .001) GO TO 20
      XFIX = 2.0
      YFIX = 2.0
   20 IF (ABS(YQSHR) .LT. .001) GO TO 25
      XFIX = 1.0
      YFIX = 1.0
   25 CONTINUE
      DO  30 I = 1,NX
      J = (I-1)*(4*NY-1)
      DO  30 K = 1,3
      CODE(J+K) = XFIX
   30 CODE(J+K+N1) = XFIX
      L = 0
      K = 1
      DO  35 I = 1,NA
      K = K + 1
      L = L + 1
      IF (L .EQ. 4) GO TO 32
      CODE(K) = YFIX
```



```
   32 IF (L .EQ. 3) GO TO 35
      CODE(K+N2) = YFIX
   35 IF (L .EQ. 4) L = 0
   37 CONTINUE
   55 IF (ABS(XLOAD) - .001) 60,102,102
   60 IF (ITYP - 2) 62,75,75
   62 IF (ABS(YLOAD) - .001)  65,110,110
   65 IF (ABS(XYLOD) - .001)  70,170,170
   70 IF (ABS(YMOM ) - .001)  75,270,270
   75 IF (ABS(XMOM ) - .001)  80,320,320
   80 IF (ITYP - 2) 82,85,85
   82 IF (ABS(XYMOM)- .001) 83,360,360
   83 IF (ABS(XQSHR)- .001) 84,500,500
   84 IF (ABS(YQSHR)- .001) 85,600,600
   85 IF (ABS(TORQ ) - .001)  800,650,650
C
C     FLAT PLATE END LOADS
C     NX
  102 PX = XLOAD*XLGT/NXND
      L = 1
      IF (ITYP-2) 105,103,103
  103 PX = XLOAD/NXND
```

```
  105 DO 109 I = 1,NX
      J = (I-1)*(4*NY-1)
      GO TO (106,107),L
  106 M = 3
      L = 2
      GO TO 108
  107 M = 1
      L = 1
  108 CONTINUE
      DO 109 K = 1,3
      FACT = 1.0
      IF (K - M) 112,111,112
  111 FACT = 2.0
      IF (I .EQ. 1) FACT = 1.0
  112 UX(J+K) = UX(J+K) - PX*FACT
  109 UX(J+K+N1) = UX(J+K+N1) + PX *FACT
      IF (IUNIT) 60,60,800
C
C     NY
  110 PY = YLOAD*YLGT/NYND
      K = 1
      L = 0
      DO 150 I = 1,NA
      K = K + 1
      L = L + 1
      IF (L-4) 120,130,120
  120 FACT = 1.0
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                      PAGE  0003
      IF (L .EQ. 2) FACT = 2.0
      IF (I .EQ. NA) FACT = 1.0
      UY(K) = UY(K) - PY*FACT
  130 IF(L-3) 140,150,140
  140 FACT = 1.0
      IF (L .EQ. 1) FACT = 2.0
      IF (I .EQ. 1) FACT = 1.0
      UY(K+N2) = UY(K+N2) + PY*FACT
  150 IF (L .EQ. 4) L = 0
      IF (IUNIT) 65,65,800
C
C     NXY
C     LOAD ON X FACE
  170 PXYA = XYLOD*XLGT/NXND
      L = 1
      DO 200 I = 1,NX
      J = (I-1)*(4*NY-1)
      GO TO (180,185),L
  180 M = 3
      L = 2
      GO TO 190
  185 M = 1
      L = 1
  190 CONTINUE
      DO 200 K = 1,3
      FACT = 1.0
      IF (K - M) 194,192,194
  192 FACT = 2.0
      IF (I .EQ. 1) FACT = 1.0
  194 UY(J+K) = UY(J+K) -  PXYA*FACT
  200 UY(J+K+N1) = UY(J+K+N1) + PXYA*FACT
C     LOAD ON Y FACE
      PXYB = XYLOD*YLGT/NYND
```

```
      K = 1
      L = 0
      DO 250 I = 1,NA
      K = K + 1
      L = L + 1
      IF (L - 4) 210,220,210
  210 FACT = 1.0
      IF (L .EQ. 2) FACT = 2.0
      IF (I .EQ. NA) FACT = 1.0
      UX(K) = UX(K) -  PXYB*FACT
  220 IF (L-3) 230,250,230
  230 FACT = 1.0
      IF (L .EQ. 1) FACT = 2.0
      IF (I .EQ. 1) FACT = 1.0
      UX(K+N2) = UX(K+N2) + PXYB*FACT
  250 IF (L .EQ. 4) L = 0
```



```
      IF (IUNIT) 70,70,800
C
C      MY
  270 PMY = YMOM*YLGT/(HT*(2*NY-1))
      K = 1
      L = 0
      DO 300 I = 1,NA
      K = K + 1
      L = L + 1
      IF (L-4) 275,280,275
  275 FACT =-1.0
      IF (L .EQ. 2) FACT = 2.0
      IF (I .EQ. NA) FACT = 1.0
      UY(K) = UY(K) -PMY*FACT
  280 IF(L-3) 290,300,290
  290 FACT =-1.0
      IF (L .EQ. 1) FACT = 2.0
      IF (I .EQ. 1) FACT = 1.0
      UY(K+N2) = UY(K+N2) +PMY*FACT
  300 IF (L .EQ. 4) L = 0
      IF (IUNIT) 75,75,800
C
C      MX
  320 CONTINUE
      IF (ITYP - 2) 325,335,335
  325 PMX = XMOM*XLGT/(HT*(2*NX-1))
      L = 1
      DO 330 I = 1,NX
      J = (I-1)*(4*NY-1)
      GO TO (326,327),L
  326 M = 3
      L = 2
      GO TO 328
  327 M = 1
      L = 1
  328 CONTINUE
      DO 330 K = 1,3
      FACT =-1.0
      IF (K - M) 331,329,331
  329 FACT = 2.0
      IF (I .EQ. 1) FACT = 1.0
  331 UX(J+K) = UX( J+K) -  PMX*FACT
  330 UX(J+K+N1) = UX(J+K+N1) +PMX *FACT
```

```
0176            GO TO 350
            C     BENDING MOMENT ON CYLINDER AND AIRFOIL
0177          335 SUM = 0.0
0178              DO 341 I = 1,NX
0179              J = (I-1)*(4*NY-1)
0180              GO TO (336,337),L

FORTRAN IV   MODEL 44  PS      VERSION 3,  LEVEL 4   DATE  71321              PAGE 0005

0181          336 M = 3
0182              L = 2
0183              GO TO 338
0184          337 M = 1
0185              L = 1
0186          338 CONTINUE
0187              DO 341 K = 1,3
0188              FACT = 1.0
0189              IF (K - M) 340,339,340
0190          339 FACT = 2.0
0191          340 CONTINUE
0192          341 SUM = SUM + Z(J+K)**2*FACT
0193              PMX = XMOM/SUM/3.0
0194              DO 348 I = 1,NX
0195              J = (I-1)*(4*NY-1)
0196              GO TO (343,344),L
0197          343 M = 3
0198              L = 2
0199              GO TO 345
0200          344 M = 1
0201              L = 1
0202          345 CONTINUE
0203              DO 348 K = 1,3
0204              FACT = 1.0
0205              IF (K - M) 347,346,347
0206          346 FACT = 2.0
0207          347 UX(J+K) = UX(J+K) + PMX*FACT*Z(I)
0208          348 UX(J+K+N1) = UX(J+K+N1) - PMX*FACT*Z(I)
0209          350 IF (IUNIT) 80,80,800
            C     MXY
0210          360 PMXYB = XYMOM*YLGT/(NT*(2*NY-1))
0211              PMXYA = XYMOM*XLGT/(NT*(2*NX-1))
            C     LOAD ON X FACES
0212              L = 1
0213              DO 400 I = 1,NX
0214              J = (I-1)*(4*NY-1)
0215              GO TO (370,375),L
0216          370 M = 3
0217              L = 2
0218              GO TO 380
0219          375 M = 1
0220              L = 1
0221          380 CONTINUE
0222              DO 400 K = 1,3
0223              FACT = 1.0
0224              IF (K - M) 390,385,390
0225          385 FACT =-2.0
0226              IF (I .EQ. 1) FACT =-1.0
0227          390 UY(J+K) = UY(J+K) -  PMXYA*FACT
```

```
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0006

        400 UY(J+K+N1) = UY(J+K+N1) + PMXVA*FACT
C           LOAD ON Y FACES
            K = 1
            L = 0
            DO 450 I = 1,NA
            K = K + 1
            L = L + 1
            IF (L - 4) 410,415,410
        410 FACT = 1.0
            IF (L .EQ. 2) FACT =-2.0
            IF (I .EQ. NA) FACT =-1.0
            UX(K) = UX(K) -  PMXYB*FACT
        415 IF (L-3) 420,450,420
        420 FACT = 1.0
            IF (L .EQ. 1) FACT =-2.0
            IF (I .EQ. 1) FACT =-1.0
            UX(K+N2) = UX(K+N2) + PMXYB*FACT
        450 IF (L .EQ. 4) L = 0
            IF (IUNIT) 83,83,800
C           QX
        500 PZ = XQSHR*XLGT/NXND
            PMY = -XQSHR    *SIDE*(NY-1)*XLGT/(2.*HT*(2*NX-1))
            L = 1
            DO 520 I = 1,NX
            J = (I-1)*(4*NY-1)
            GO TO (505,507),L
        505 N = 3
            L = 2
            GO TO 510
        507 N = 1
            L = 1
        510 CONTINUE
            DO 520 K = 1,3
            FACT = - 1.0
            IF (K-N) 517,515,517
        515 FACT = 2.0
            IF (I .EQ. 1) FACT = 1.0
        517 UX(J+K) = UX(J+K) - PMY*FACT
            UX(J+K+N1) = UX(J+K+N1)- PMY*FACT
            UZ(J+K) = UZ(J+K) + PZ*ABS(FACT)
        520 UZ(J+K+N1) = UZ(J+K+N1) - PZ*ABS(FACT)
            IF (IUNIT) 84,84,800
C           QY
        600 PZ = YQSHR*YLGT/NYND
            PMX = -YQSHR    *(XLGT-Y2)*YLGT/(2.*HT*(2*NY-1))
            K = 1
            L = 0
            DO 620 I = 1,NA
            K = K + 1
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0007

            L = L + 1
            IF (L-4) 605,610,605
        605 FACT = -1.0
            IF (L .EQ. 2) FACT = 2.0
            IF (I .EQ. NA) FACT = 1.0
            UY(K) = UY(K) - PMX*FACT
            UZ(K) = UZ(K) + PZ*ABS(FACT)
        610 IF (L-3) 615,620,615
```

```
  615 FACT = -1.0
      IF (L .EQ. 1) FACT = 2.0
      IF (I .EQ. 1) FACT = 1.0
      UY(K+N2) = UY(K+N2) - PMX*FACT
      UZ(K+N2) = UZ(K+N2) - PZ*ABS(FACT)
  620 IF (L .EQ. 4) L = 0
      IF(IUNIT) 85,85,800
C
C     GENERATE TORSION LOADS ON CYLINDER
  650 IF (ITYP - 2) 800,660,720
  660 RAD1 = SQRT(Y(1)**2+Z(1)**2)
      RAD2 = RAD1 + HT
      PA = TORQ/(2.*RAD1*NX+2.*RAD2*NX+8.*(RAD1+RAD2)/2.*NX*4.)
      PB = RAD2/RAD1*PA
      DO 690 I = 1,NX
      J = (I-1)*(4*NY-1)
      THETA = ATAN2(Z(J+K),Y(J+K))
      SINTH = SIN(THETA+1.57079 )
      COSTH = COS(THETA+1.57079)
      GO TO (685,686),L
  685 M = 3
      L = 2
      GO TO 687
  686 M = 1
      L = 1
  687 CONTINUE
      DO 690 K = 1,3
      FACT = PA
      IF (K - M) 689,688,689
  688 FACT = 2.0*PB
  689 UY(J+K) = UY(J+K) +     COSTH*FACT
      UZ(J+K) = UZ(J+K) +     SINTH*FACT
      UY(J+K+N1) = UY(J+K+N1) -     COSTH*FACT
  690 UZ(J+K+N1) = UZ(J+K+N1) -     SINTH*FACT
      GO TO 800
C
C     GENERATE TORSION LOADS ON AIRFOIL
  720 NEND = 5*NY/4 + 1
      ARM = 0.0
      DO 750 I = 1,NX
      J = (I-1)*(4*NY-1)
```



```
      GO TO (745,746),L
  745 M = 3
      L = 2
      GO TO 747
  746 M = 1
      L = 1
  747 CONTINUE
      DO 750 K = 1,3
      FACT = 1.0
      IF (K - M) 749,748,749
  748 FACT = 2.0
  749 ANG = SLOPA(J+K)
      IF (Z(J+K) .LT. 0.0) ANG = ANG - 3.14159
  750 ARM = ARM + (Z(J+K)*COS(ANG) - Y(J+K)*SIN(ANG))*FACT
      PA = TORQ/ARM/3.0
      TORK = 0.0
      DO 760 I = 1,NX
      J = (I-1)*(4*NY-1)
      GO TO (755,756),L
```

```
 755 M = 3
     L = 2
     GO TO 757
 756 M = 1
     L = 1
 757 CONTINUE
     DO 760 K = 1,3
     FACT = 1.0
     IF (K - M) 759,758,759
 758 FACT = 2.0
 759 ANG = SLOPA(J+K)
     IF (Z(J+K) .LT. 0.0) ANG = ANG - 3.14159
     UY(J+K) = UY(J+K) + PA*COS(ANG)*FACT
     UZ(J+K) = UZ(J+K) + PA*SIN(ANG)*FACT
     TORK = TORK + (Z(J+K)*UY(J+K) - Y(J+K)*UZ(J+K))*FACT
     UY(J+K+N1) = UY(J+K+N1) - PA*COS(ANG)*FACT
 760 UZ(J+K+N1) = UZ(J+K+N1) - PA*SIN(ANG)*FACT
     WRITE (6,5020) I ,TORK,ARM,PA
 800 CONTINUE
     IF (ITYP-2) 810,830,830
 810 NFIX = N2
     DO 820 I = 1,NY
     NFIX = NFIX + 4
  54 CODE(NFIX) = 7.0
     UX(NFIX) = 0.0
     UY(NFIX) = 0.0
 820 UZ(NFIX) = 0.0
 830 CONTINUE
6020 FORMAT ( I10,6F15.5 )
C
```

TAPE 3


```
 850 WRITE (3) (UX(I),UY(I),UZ(I),I=1,NUMNP)
     IF (IHOLE .EQ. 0) GO TO 860
     CODE(IHOLE-1) = 7.0
     CODE(IHOLE  ) = 7.0
     CODE(IHOLE+1) = 7.0
 860 CONTINUE
     NUM = 4
     IF (NY .LT. 4) NUM = 0
     NOD1 = 4*NY + NUM
     CODE(NOD1) = 7.0
     CODE(NOD1+4) = 6.0
     CODE(NOD1+4*NY+1) = 3.0
     WRITE (3) (CODE(I),I=1,NUMNP)
     IF (ITEST.NE.0 )
    XWRITE(6,30007)  (( I,CODE(I)),I=1,NUMNP)
30007 FORMAT(I8,F7.1)
     RETURN
     END
```


COMMON BLOCK / COMN   / MAP SIZE 00006C

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	000000	NUMEL	000004	NLOD	000008	NLIN	00000C	MBAND	000010
NUMBLK	000014	NBN	000018	NEL	00001C				

COMMON BLOCK / ONE / MAP SIZE 0000E8

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	000000	HT	000004	THK	000008	THETA	00001C	Y1OFF	000020
Y2OFF	000024	THT	000028	DUM	00002C	TC	000030	XC	00005C
HED	000088	NX	000088	NY	00008C	IOPT	0000C0	ITYP	0000C4
IFACE	0000C8	IHOLE	0000CC	ITET	0000D0	ICNT	0000D4	NNEW	0000D8

COMMON BLOCK / LODA / MAP SIZE 000024

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
XLOAD	000000	YLOAD	000004	XYLOD	000008	XMOM	00000C	YMOM	000010
XYMOM	000014	TORQ	000018	XQSHR	00001C	YQSHR	000020		

COMMON BLOCK / LODS / MAP SIZE 0012C0

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
CODE	000000	UX	000480	UY	000960	UZ	000E10		

COMMON BLOCK / DUMP / MAP SIZE 000004

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
ITEST	000000								

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
I	0003BC	N1	0003C0	N2	0003C4	NA	0003C8	Y2	0003CC
NXND	0003D0	NYND	0003D4	XLGT	0003D8	YLGT	0003DC	N4	0003E0
KONST	0003E4	IUNIT	0003E8	XFIX	0003EC	YFIX	0003F0	J	0003F4
K	0003F8	L	0003FC	PX	000400	M	000404	FACT	000408
PY	00040C	PXYA	000410	PXYB	000414	PMY	000418	PMX	00041C
SUM	000420	PMXYB	000424	PMXYA	000428	PZ	00042C	RAD1	000430
RAD2	000434	PA	000438	PB	00043C	SINTH	000440	COSTH	000444
NEND	000448	ARM	00044C	ANG	000450	TORK	000454	NFIX	000458
NUM	00045C	NOD1	000460						

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
X	000464	Y	000468	Z	00046C	SLOPA	000470		

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	000474	SQRT	000478	ATAN2	00047C	SIN	000480	COS	000484

LABEL MAP

LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION

40	0005A8	50	000602	20	00079E	25	0007C6	30	00081E
32	0008BE	35	0008EE	37	00091E	55	00091E	60	000938
62	000950	65	00096A	70	000984	75	00099E	80	0009B8
82	0009D0	83	0009EA	84	000A04	85	000A1E	102	000A3E
103	000A8C	105	000ABA	106	000B14	107	000B2E	108	000B42
111	000B6A	112	000B8C	109	000BCA	110	000C40	120	000C86
130	000D10	140	000D24	150	000D9C	170	000DDE	180	000E70
185	000E8A	190	000E9E	192	000EC6	194	000EE8	200	000F26
210	001000	220	00105A	230	00106E	250	0010E6	270	001128
275	0011AE	280	001208	290	00121C	300	001294	320	0012D6
325	0012EE	326	001390	327	00134A	328	0013BE	329	0013E6
331	001408	330	001446	335	001480	336	001514	337	00152E
338	001542	339	00156A	340	001576	341	001576	343	001638
344	001652	345	001666	346	00168E	347	00169A	348	0016EA
350	001754	360	001766	370	00183C	375	001856	380	00186A
385	001892	390	0018B4	400	0018F2	410	00199A	415	0019F4

```
  420  001A08     450  001A80     500  001AC2     505  001B8C     507  001BD6
  510  001BEA     515  001C12     517  001C34     520  001CE5     600  001D60
  605  001E20     610  001E8C     615  001EA0     620  001F52     650  001F94
  660  001F82     685  002170     686  00218A     687  00219E     688  0021C2
  689  0021D2     690  002280     720  0022EC     745  00236C     746  002386
  747  00239A     748  0023C2     749  0023CE     750  002420     755  002520
  756  00253A     757  00254E     758  002576     759  002582     760  002730
  800  0027D8     810  0027F0      54  002814     820  00283C     830  00285A
 6020  00285A     850  00286C     860  00291A   30007  002A58

 TOTAL MEMORY REQUIREMENTS 002B04 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//STRSS  EXEC FORTRAN(BCD,MAP)

 FORTRAN IV    MODEL 44   PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0001

              C      ************************ OVERLAY B ****************************************
 0001                SUBROUTINE STRSS
 0002                CALL LOAD('TETRA9')
 0003                CALL STRESS
              C      FORM STIFFNESS MATRIX                                        PSMN1020
 0004                CALL LOAD('TETRA10')
 0005                CALL STIFF
 0006                RETURN
 0007                END

 FORTRAN IV    MODEL 44   PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0002

                                  SUBPROGRAMS CALLED
SYMBOL     LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
LOAD       0000D0       STRESS    0000D4       STIFF     0000D8

 TOTAL MEMORY REQUIREMENTS 000190 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//STRESS EXEC FORTRAN(BCD,MAP)

 FORTRAN IV    MODEL 44   PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0001

              C************************************* OVERLAY BA ********************************
 0001                SUBROUTINE STRESS
              C                                                                   PSMN0010
 0002                COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
 0003                COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
                   1  ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
 0004                COMMON / XYZ / X(300),Y( 300),Z(300),UX(600),UY(600),UZ(600)
                   1  , CODE(600),IDO(600),IELM
 0005                COMMON / ELM / S(18,18),LM(6),T,ETR,EPRM,ELGT,G,ST(3,18,4),INTL
 0006                COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLD,NR1,NR2
 0007                DIMENSION ID(300),          NI(300),IN(300),ML(3),MM(6)
 0008                DIMENSION DUM1(600),DUM2(600),DUM3(600),DUM4(600),B(1)
 0009                EQUIVALENCE (LM(4),IS),(LM(3),IR),(LM(2),IQ),(LM(1),IP)
                   $,(MM(6),LU),(MM(5),LT),(MM(4),IS),(MM(3),LR),(MM(2),LQ)
                   $,(MM(1),LP)
 0010                EQUIVALENCE(ML(3),JR),(ML(2),JQ),(ML(1),JP),(B(1),UX(1))
              C
 0011                REWIND 3                                                     TAPE 3
```

```
      NTAPE =3
      IF (ICNT-1) 45,45,50
   45 WRITE (6,6010)      NUMNP,NUMEL,NLOD
 6010 FORMAT (1H1  ///
     1 30H- NUMBER OF NODAL POINTS------ I4 /
     2 30H0 NUMBER OF ELEMENTS---------- I4 /
     3 30H0 NUMBER OF LOAD CASES      --- I4 )
C       READ NODAL DATA FROM TRANSFER TAPE
C     READ AND PRINT OF NODAL POINT DATA
   50 READ  (3) (ID(I),X(I),Y(I),Z(I),I=1,NUMNP)                  TAPE 3
      IF (ICNT-1) 80,80,120
   80 MPRINT=0
      DO 110 N=1,NUMNP
      IF(MPRINT) 90,85,90
   85 WRITE(6,6020)
 6020 FORMAT (6H1NODAL, /, 6H POINT, 4X, 1HX , 8X, 1HY, 8X, 1HZ,// )
      MPRINT=50
   90 MPRINT=MPRINT-1
   95 WRITE(6,6030)ID(N),        X(N),Y(N),Z(N)
 6030 FORMAT (I4, 4F10.4,3F15.4 )
  110 CONTINUE
C        READ AND PRINT OF ELEMENT PROPERTIES
C         BEGIN BOOK-KEEPING ON ADDITIONAL NODES AT MID-POINTS OF SIDES
  120 KA1=0
      DO 300  I=1,NUMEL
      READ (3)IDP,IP,IQ,IR,IS,ETR,ELGT,EPRR,G,T,IMTL               TAPE 3
      JQ = IQ
      JP = IP
      IF (IMTL - 3) 200,200,290
  200 IF (I .GT. 1) GO TO 220
      IF (IS .GT. 0) GO TO 215
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0002

```
  210 NI(1)=NID(IQ,IR)
      IN(1)=MAXO(IQ,IR)
      GO TO 218
  215 NI(1) = NID(IQ,IS)
      IN(1) = MAXO(IQ,IS)
  218 JP=NI(1)
      NI(2)=NID(IR,IP)
      IN(2)=MAXO(IR,IP)
      JQ = NI(2)
      NADD = 2
      GO TO 280
  220 CONTINUE
      DO 270  L=1,2
      IF (L .EQ. 2) GO TO 240
      KA8 = IQ
      KA9 = IR
      IF (IS .GT. 0) KA9 = IS
      GO TO 245
  240 KA8 = IR
      KA9 = IP
  245 KA5 = NID(KA8,KA9)
      KA7 = MAXO(KA8,KA9)
      DO 250  M = 1,NADD
      IF (KA5-(NI(NADD+1-M))) 250,260,250
  250 CONTINUE
      NADD = NADD + 1
      NI(NADD) = KA5
      IN(NADD) = KA7
```

```
0064        260 CONTINUE
0065            ML(L) = KAS
0066        270 CONTINUE
0067        280 CONTINUE
0068        290 CONTINUE
0069            NR2=I
0070            WRITE(2*NR2)
              1          IDP,IP,IQ,IR,IS,JP,JQ,ETR,ELGT,EPRM,G,T,INTL
0071       6060 FORMAT(12I6 )
0072        300 CONTINUE
0073            NODES = NUMNP + NADD
0074            NRS=3*NODES
0075            DO 415 K = 1,NLOD
0076            MPRINT=0
0077            READ  (3) (UX(I),UY(I),UZ(I),I=1,NUMNP)                   TAPE 3
0078            READ (3) (CODE(I),I=1,NUMNP)                              TAPE 3
           C         RE-ORDER LOAD MATRIX TO ACCOUNT FOR ADDITIONAL NODES USING
           C        MODIFY BOUNDARY VALUES TO INCLUDE ADDITIONAL NODES
0079            DO 350 I = 1,NODES
0080            DUM1(I) = 0.0
0081            DUM2(I) = 0.0
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0003

```
0082            DUM3(I) = 0.0
0083        350 DUM4(I) = 0.0
0084            IDO(I)=ID(I)
0085            DUM1(I) = UX(I)
0086            DUM2(I) = UY(I)
0087            DUM3(I) = UZ(I)
0088            DUM4(I) = CODE(I)
0089            KA7 = 1
0090            DO 400 I = 1,NUMNP
0091            DO 370 J = 1,NADD
0092            IF (ID(I)-IN(J)) 370,360,370
0093        360 KA7 = KA7 + 1
0094            IDO(KA7) = NI(J)
0095            ID1 = IDO(KA7)/1000
0096            ID2 = IDO(KA7) - ID1*1000
0097            IF (UX(ID1)) 362,365,362
0098        362 IF (UX(ID2)) 364,365,364
0099        364 IF (UX(ID1)/UX(ID2) .LE. 0.0) GO TO 365
0100            ALOD = ABS(UX(ID1))
0101            BLOD = ABS(UX(ID2))
0102            CLOD = AMIN1(ALOD,BLOD)
0103            DUM1(KA7) = 4.*CLOD*UX(ID1)/ALOD
0104        365 IF (UY(ID1)) 366,368,366
0105        366 IF (UY(ID2)) 367,368,367
0106        367 IF (UY(ID1)/UY(ID2) .LE. 0.0) GO TO 368
0107            ALOD = ABS(UY(ID1))
0108            BLOD = ABS(UY(ID2))
0109            CLOD = AMIN1(ALOD,BLOD)
0110            DUM2(KA7) = 4.*CLOD*UY(ID1)/ALOD
0111        368 IF (UZ(ID1)) 371,374,371
0112        371 IF (UZ(ID2)) 373,374,373
0113        373 IF (UZ(ID1)/UZ(ID2) .LE. 0.0) GO TO 374
0114            ALOD = ABS(UZ(ID1))
0115            BLOD = ABS(UZ(ID2))
0116            CLOD = AMIN1(ALOD,BLOD)
0117            DUM3(KA7) = 4.*CLOD*UZ(ID1)/ALOD
0118        374 CONTINUE
0119        370 CONTINUE
```

```
      IF (I-NUMNP) 380,390,380
  380 KA7 = KA7 + 1
      IDD(KA7) = ID(I+1)
      DUM1(KA7) = UX(I+1)
      DUM2(KA7) = UY(I+1)
      DUM3(KA7) = UZ(I+1)
      DUM4(KA7) = CODE(I+1)
  390 CONTINUE
  400 CONTINUE
      DO 410 I = 1,NODES
      J=3*I
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4    DATE  71321                    PAGE 0004
      B(J-2)= DUM1(I)
      B(J-1)= DUM2(I)
      B(J)  = DUM3(I)
      IF (ICNT.GT.1) GO TO 410
      IF(MPRINT.NE.0) GO TO 406
      MPRINT=50
      WRITE (6,6065) K
 6065 FORMAT ( 1H1, 20X, 9HLOAD CASE, I4,/, 5X, 3HNEW, 2X, 3HOLD,/,
     1  4X, 4HNODE, 2X, 4HNODE, 10X, 2HUX, 11X, 2HUY, 11X, 2HUZ,// )
  406 CONTINUE
      MPRINT=MPRINT-1
      WRITE(6,6070) I,IDD(I),B(J-2),B(J-1),B(J)
 6070 FORMAT(2I6, 3F15.4 )
  410 CONTINUE
      DO 412 I = 1,NUMNP
  412 DUM1(I) = CODE(I)
      DO 413 I = 1,NODES
  413 CODE(I) = DUM4(I)
  415 CONTINUE
      DO 480 I = 1,NODES
  417 IF (IDD(I) - 1000) 480,480,420
  420 KA1 = IDD(I)/1000
      KA2=(IDD(I)- KA1*1000)
      KT=0
      DO 470 J = 1,NUMNP
      IF (ID (J)-KA1) 440,430,440
  430 N1 = J
      GO TO 460
  440 IF (ID (J) - KA2) 460,450,460
  450 N2 = J
  460 CONTINUE
  470 CONTINUE
      COD1 = DUM1(N1)
      COD2 = DUM1(N2)
      CODE(I) = AMIN1(COD1,COD2)
  480 CONTINUE
      REWIND 3                                                      TAPE 3
      KK = 0
      MPRINT=0
      DO 720 I = 1,NUMEL
      NR2=I
      READ(2'NR2)
     1          IDP,KP,KQ,KR,KS,JP,JQ,ETR,ELBT,EPRM,G,T,INFL        TAPE 2
      IS = 0
      LU = 0
      IELM = 1
      DO 560 J = 1,NUMNP
      IF (KP-ID(J)) 510,500,510
  500 IP = J
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0005

            GO TO 550
      510 IF (KQ-ID(J)) 530,520,530
      520 IQ = J
            GO TO 550
      530 IF (KR-ID(J)) 545,540,545
      540 IR = J
            GO TO 550
      545 IF (KS-ID(J)) 550,547,550
      547 IS = J
            IELM = 2
      550 CONTINUE
      560 CONTINUE
            DO 670 J = 1,NODES
            IF (KP-IDD(J)) 580,570,580
      570 LP = J
            GO TO 660
      580 IF (KQ-IDD(J)) 600,590,600
      590 LQ = J
            GO TO 660
      600 IF (KR-IDD(J)) 620,610,620
      610 LR = J
            GO TO 660
      620 IF (JP-IDD(J)) 640,630,640
      630 LS = J
            GO TO 660
      640 IF (JQ-IDD(J)) 652,650,652
      650 LT = J
            GO TO 660
      652 IF (KS-IDD(J)) 660,655,660
      655 LA = J
      660 CONTINUE
      670 CONTINUE
            LU = LT
            IF (IELM .EQ. 1) GO TO 675
            LT = LS
            LS = LA
      675 IF (INTL .LE. 3) GO TO 690
      680 LS = LQ
            LT = LQ
            LU = LQ
      690 IF (NLIN-1) 692,695,695
      692 IF (ICNT-1) 695,695,697
      695 IF(NPRINT) 170,165,170
      165 WRITE(6,6040)
     6040 FORMAT(0H1ELEMENT,                     /TH NUMBER,
         1 2X,1HJ,3X,1HK,3X,1HL,6X, 2HEX, 9 X, 2HEY, 10X, 3HEXY,12X
         1  , 1HG,  9X, 1HT,   5X,23H LP  LQ  LR  LS  LT  LU  , // )
            NPRINT=50
      170 NPRINT=NPRINT-I

FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0006

            WRITE (6,6050) IDP,IP,IQ,IR,IS,ELGT,ETR,EPRM,G,T,LP,LQ,LR,LS,LT,LU
     6050 FORMAT (I6,4I4 , 4F12.0,F10.4,6I4 )
      697 CONTINUE
C                  STORE ELEMENTAL DATA ON  TAPE 1
            NR1=I
            WRITE(1'NR1) IDP,KP,KQ,KR,KS,LM,MM,ETR,ELGT,EPRM,G,T,INTL,IELM     TAPE 1
C              COMPUTE BAND WIDTH
            IF (LU .EQ. 0) LU = LT
            NMAX = MAXO(LP,LQ,LR,LS,LT,LU)
            NMIN = MINO(LP,LQ,LR,LS,LT,LU)
```

```
0233          IF (KK-(NMAX-NMIN)) 700,700,710
0234      700 KK = NMAX - NMIN
0235      710 CONTINUE
0236      720 CONTINUE
        C     NOR1 IS THE NUMBER OF RECORDS REQUIRED BY FILE 1 AND 2
0237          NOR1=NUMEL
0238          WRITE(6,1) NOR1
0239        1 FORMAT(1H ,I5,' IS THE NUMBER OF RECORDS REQUIRED BY FILE 1 AND 2'
             1)
0240          MBAND =3*(KK+1)
        C     NOBYT IS THE NUMBER OF BYTES PER RECORD REQUIRED BY FILE 4
        C     NOBYT=BANDWIDTH OF MATRIX * 4 BYTES PER WORD * 3 ROWS IN MATRIX
0241          NOBYT=MBAND*4*3
0242          WRITE(6,2) NOBYT
0243        2 FORMAT(1H ,I6,' IS THE NUMBER OF BYTES PER RECORD REQUIRED BY FILE
             1 4')
0244          NBAND = MBAND/3
0245          IF (ICNT .EQ. 1) WRITE (6,6100) NBAND
0246     6100 FORMAT ( 1H1 , 5X, 15H1/2 BANDWIDTH # , I5 )
        C     ADD CONCENTRATED FORCES
0247          M=MBAND-1
0248          NR4STF=1
0249          NRECST = (3*M)/NREC
0250          IF(NREC*NRECST.LT.3*M )  NRECST=NRECST+1
0251          NRECLD = NRS/NREC
0252          IF (NRECLD*NREC.LT.NRS ) NRECLD=NRECLD +1
0253          NR4DIA = NODES*NRECST+1
0254          NR4LOD = NR4DIA + NRECLD
0255          NR4 = NR4LOD
0256          DO 800 I = 1,NLOD
0257      800 CALL OUTIN(1,B,NRS)
        C     NR4 IS THE NUMBER RECORDS REQUIRED BY FILE 4
0258          WRITE(6,3) NR4
0259        3 FORMAT(1H ,I5,' IS THE NUMBER OF RECORDS REQUIRED BY FILE 4')
0260          NUMNP = NODES
0261          RETURN
0262          END
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0007

COMMON BLOCK / COMN / MAP SIZE 00006C

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	000000	NUMEL	000004	NLOD	000008	NLIN	00000C	MBAND	000010
NUMBLK	000014	NBN	000018	NEL	00001C				

COMMON BLOCK / ONE / MAP SIZE 0000E8

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	000000	HT	000004	THK	000008	THETA	00001C	Y1OFF	000020
Y2OFF	000024	THT	000028	DUM	00002C	TC	000030	XC	00005C
HED	000088	NX	000088	NY	00008C	IOPT	0000C0	ITYP	0000C4
IFACE	0000C8	IHOLE	0000CC	ITET	0000D0	ICNT	0000D4	NNEW	0000D8

COMMON BLOCK / XYZ / MAP SIZE 003CF4

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
X	000000	Y	0004B0	Z	000960	UX	000E10	B	000E10
UY	001770	UZ	0020D0	CODE	002A30	IDO	003390	IELM	003CF0

COMMON BLOCK / ELM / MAP SIZE 0008A0

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
S	000000	LM	000510	IP	000510	IQ	000514	IR	000518
IS	00051C	T	000528	ETR	00052C	EPRM	000530	ELGT	000534
G	000538	ST	00053C	INTL	00089C				

```
                                   COMMON BLOCK / RAX     / MAP SIZE  000024
 SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
 NR4       000000       NR4LOD    000004       NR4STF    000008       NR4DIA    00000C       NREC      000010
 NRECST    000014       NRECLD    000018       NR1       00001C       NR2       000020

                                   EQUIVALENCE DATA MAP
 SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
 MM        0002C0       LP        0002CC       LQ        0002C4       LR        0002C8       LS        0002CC
 LT        0002D0       LU        0002D4       ML        0002D8       JP        0002D8       JQ        0002DC
 JR        0002E0

                                   SCALAR MAP
 SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
 NTAPE     0002E4       I         0002E8       MPRINT    0002EC       N         0002F0       KA1       0002F4
 IDP       0002F8       NADD      0002FC       L         000300       KA8       000304       KA9       000308
 KA5       00030C       KA7       000310       M         000314       NODES     000318       NRS       00031C
 K         000320       J         000324       ID1       000328       ID2       00032C       ALDD      000330
 BLOD      000334       CLOD      000338       KA2       00033C       KT        000340       N1        000344
 N2        000348       COD1      00034C       COD2      000350       KK        000354       KP        000358
 KQ        00035C       KR        000360       KS        000364       LA        000368       NMAX      00036C
 NMIN      000370       NOR1      000374       NOBYT     000378       NBAND     00037C

                                   ARRAY MAP
 SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION

FORTRAN IV   MODEL 44  PS      VERSION 3,  LEVEL 4   DATE  71321                         PAGE  0008

ID         000380       N1        000830       IN        000CE0       DUM1      001190       DUM2      001AF0
DUM3       002450       DUM4      002D80

                                   SUBPROGRAMS CALLED
SYMBOL     LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
IBCOM#     003710       NIO       003714       OUTIN     003718       MAXO      00371C       AMIN1     003720
MINO       003724

                                   LABEL MAP
LABEL      LOCATION     LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION
    45     003854        6010     00388C          50     003902          80     003988          85     0039A8
  6020     0039C0          90     0039F6          95     003A06        6030     003A50         110     003A66
   120     003A7C         200     003B24         210     003B48         215     003B6E         218     003B8E
   220     003BD0         240     003C1E         245     003C32         250     003C84         260     003CBE
   270     003CCE         280     003CE4         290     003CE4        6060     003D78         300     003D86
   350     003EBC         360     003F56         362     003FBC         364     003FD4         365     004048
   366     004068         367     004088         368     004100         371     004118         373     004130
   374     0041A4         370     0041A4         380     0041CE         390     004232         400     004232
  6065     0042EC         406     00433A        6070     0043A4         410     0043B6         412     0043D4
   413     004416         415     004448         417     004466         420     004486         430     0044E0
   440     0044EE         450     004506         460     00450E         470     00450E         480     00457A
   500     00468C         510     00469E         520     004686         530     0046C8         540     0046E0
   545     0046F2         547     00470A         550     004726         560     004726         570     004764
   580     004772         590     00478E         600     00479C         610     0047B8         620     0047C6
   630     0047E2         640     0047F0         650     00480C         652     004814         655     004836
   660     00483E         670     00483E         675     00487E         680     004894         690     0048AC
   692     0048C4         695     0048DC         165     0048E8        6040     004900         170     004972
  6050     004A20         697     004A3A         700     004830         710     004B3C         720     004B3C
     1     004B84           2     004C04        6100     004C0C         800     004D72           3     004DC4

  TOTAL MEMORY REQUIREMENTS 004E60 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//STIFF  EXEC FORTRAN(BCD,MAP)
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0001

      C********************************** OVERLAY BB ********************************
            SUBROUTINE STIFF
      C                                                                 PSTF0010
            COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
            COMMON / XYZ / X(300),Y(300),Z(300),UX(600),UY(600),UZ(600)
           1  , CODE(600),IDD(600),IELM
            COMMON / ELM / S(18,18),LM(6),T,ETR,EPRM,ELGT,G,ST(3,18,4),INTL
            COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLD,NR1,NR2
            COMMON /DUMP / NTEST
            DIMENSION    A(2700 ),DIAG(1800)
            DIMENSION MM(6)
      C     INITIALIZATION
            REWIND 3                                                    TAPE 3
            NR4=NR4STF
            NRS=3*NUMNP
            M=MBAND-1
            NE=3*M
            DO 49 I=1,NE
       49   A(I)=0.
            DO 50 I=1,NRS
         50  DIAG(I)=0.0
            DO 51 J=1,NUMNP
            CALL OUTIN (1,A      ,NE    )
         51 CONTINUE
            DO 1000 N=1,NUMEL
            NR1=N
            READ(1'NR1) IDP,KP,KQ,KR,KS,LM,MM,ETR,ELGT,EPRM,G,T,INTL,IELM     TAPE 1
            IF (INTL .GT. 3) GO TO 62
            IF (IELM .EQ. 2) GO TO 64
            I1 = MM(3)
            MM(3) = MM(2)
            MM(2) = I1
            I2 = MM(5)
            MM(5) = MM(4)
            MM(4) = I2
            NNOD = 5
            GO TO 65
         62 NNOD = 3
            GO TO 65
         64 NNOD = 6
            I1 = MM(5)
            MM(5) = MM(6)
            MM(6) = I1
       65   CALL TRIM6
       1007 FORMAT (1H ,7I10/ (1H ,1P15E8.1))
            IF (NTEST .EQ. 0) GO TO 557
            NOD3 = 3*NNOD
            WRITE (6,30009) MM
            DO 556 I = 1,NOD3

FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0002

        556 WRITE (6,6000) (S(I,J),J=1,NOD3)
        557 CONTINUE
       6000 FORMAT ( 10F12.0 )
      30009 FORMAT(12I6)
            WRITE(3)IDP,ST,KP,KQ,KR,KS,LM,ETR,ELGT,EPRM,G,T,INTL,MM,NNOD      TAPE 3
            DO 400I=1,NNOD
        400 LM(I)=3*MM(I)-3
            DO 820 I=1,NNOD
            NR4=(MM(I)-1)*NRECST+1
```

```
      LR4=NR4
      CALL OUTIN (2,A,NE)
      NR4=LR4
      DO 700 K=1,3
      NS=M*(K-1)
      II=LM(I)+K
      KK=3*I-3+K
      DO 800 J=1,NNOD
 803    DO 805 L= 1,3
      JJ=LM(J)+L
      IF( JJ.GE.II) GO TO 804
      JJ=JJ-II+MBAND+NS
      LL=3*J-3+L
 805  A(JJ    )=A(JJ    )+S(KK,LL)
 804  CONTINUE
 800  CONTINUE
         DIAG(II)=DIAG(II)+S(KK,KK)
 700  CONTINUE
      CALL OUTIN (1,A,NE)
 820  CONTINUE
1000  CONTINUE
      IF(NTEST.EQ.0) GO TO 1011
      WRITE(6,1005)
1005  FORMAT(24H MERGED STIFFNESS MATRIX  )
      NR4=NR4STF
      DO 1010 I=1,NRS,3
      CALL OUTIN (2,A,NE)
      WRITE(6,1008) I,(A(K   ),K=1,NE),DIAG(I),DIAG(I+1),DIAG(I+2)
1008  FORMAT(1H ,4HROW#,I5,3X,10E12.4/(1H ,11E12.4))
1010  CONTINUE
1011  CONTINUE
C     INSERT B V .S AND SET UP DIAG TAPE
      DO 2000 I=1,NUMNP
      IF(CODE(I)-.999)2001,2002,2002
2002  CODE(I)=CODE(I)+0.00001
      NJ=1
      NC=CODE(I)
      GO TO (1,2,3,4,5,6,7),NC
   1    NS=1
      NE=1

FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                PAGE  0003

      GOTO 8
   2    NS=2
      NE=2
      GO TO 8
   3    NS=3
      NE=3
      GO TO 8
   4    NS=1
      NE=2
      GO TO 8
   5    NS=1
      NE=3
      NJ=2
      GO TO 8
   6    NS=2
      NE=3
      GO TO 8
   7    NS=1
      NE=3
```

```
    8   CONTINUE
        DO 9 K=NS,NE,NJ
    9     DIAG(3*I-3+K)=DIAG(3*I-3+K)*1.0E+12  +1.0 E+12
 2001 CONTINUE
 2000 CONTINUE
      NR4=NR4DIA
      CALL OUTIN (1,DIAG  ,NRS  )
      IF(NTEST.EQ.0) GO TO 1020
      WRITE(6,1019)      (            DIAG(I),I=1,NRS)
 1019 FORMAT(                          1H ,11E12.4)
 1020 CONTINUE
      WRITE (3) (IDO(I),I=1,NUMNP)                              TAPE 3
      RETURN
      END
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0004

COMMON BLOCK / COMN / MAP SIZE 00006C

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	000000	NUMEL	000004	NLOC	000008	NLIN	00000C	MBAND	000010
NUMBLK	000014	NBN	000018	NEL	00001C				

COMMON BLOCK / XYZ / MAP SIZE 003CF4

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
X	000000	Y	0004B0	Z	000960	UX	000E10	UY	001770
UZ	002000	CODE	002A30	IDO	003390	IELM	003CF0		

COMMON BLOCK / ELM / MAP SIZE 0008A0

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
S	000000	LM	000510	T	000528	ETR	00052C	EPRM	000530
ELGT	000534	G	000538	ST	00053C	INTL	00089C		

COMMON BLOCK / RAX / MAP SIZE 000024

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NR4	000000	NR4LOD	000004	NR4STF	000008	NR4DIA	00000C	NREC	000010
NRECST	000014	NRECLD	000018	NR1	00001C	NR2	000020		

COMMON BLOCK / DUMP / MAP SIZE 000004

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NTEST	000000								

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NRS	0001B0	M	0001B4	NE	0001B8	I	0001BC	J	0001C0
N	0001C4	IDP	0001C8	KP	0001CC	KQ	0001D0	KR	0001D4
KS	0001D8	I1	0001DC	I2	0001E0	NNOD	0001E4	NOD3	0001E8
LR4	0001EC	K	0001F0	NS	0001F4	II	0001F8	KK	0001FC
L	000200	JJ	000204	LL	000208	NJ	00020C	NC	000210

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
A	000214	DIAG	002C44	MM	004864				

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
IBCOM#	00487C	OUTIN	004880	TRIM6	004884				

```
                                        LABEL MAP
LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION
   49     00495C          50     00498A          51     0049CE          62     004AFA          64     004B0C
   65     004534        1007     004B3E         556     004BAC         557     004C26        6000     004C26
30009     004C34         400     004CEC         803     004DC8         805     004E2A         804     004E66
  800     004E66         700     004E82         820     004ED2        1000     004EE8        1005     004F2C
 1008     004FF0        1010     005016        1011     005028        7002     005058           1     0050E0

 FORTRAN IV    MODEL 44   PS        VERSION 3,  LEVEL 4   DATE  71321                       PAGE  0005

    2     0050FA           3     005114           4     00512E           5     005148           6     00516A
    7     005184           8     005198           9     0051A0        2001     005206        2000     005206
 1019     0052A4        1020     005286

 TO    MEMORY REQUIREMENTS 005364 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//NID     EXEC FORTRAN(BCO,MAP)

 FORTRAN IV    MODEL 44   PS        VERSION 3,  LEVEL 4   DATE  71321                       PAGE  0001

 0001              FUNCTION NID(KP,KQ)
 0002              NID = MAXO(KP,KQ)*1000 + MINO(KP,KQ)
 0003              RETURN
 0004              END

 FORTRAN IV    MODEL 44   PS        VERSION 3,  LEVEL 4   DATE  71321                       PAGE  0002

                                        EQUIVALENCE DATA MAP
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
NID       000004

                                        SCALAR MAP
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
KP        0000D8       KQ        0000DC

                                        SUBPROGRAMS CALLED
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
MAXO      0000E0       MINO      0000E4

 TOTAL MEMORY REQUIREMENTS 0001E0 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//TRIM6   EXEC FORTRAN(BCO,MAP)
 FORTRAN IV    MODEL 44   PS        VERSION 3,  LEVEL 4   DATE  71321                       PAGE  0001

 0001              SUBROUTINE TRIM6
 0002              COMMON / XYZ / X(300),Y(300),Z(300),UX(600),UY(600),UZ(600)
                  1  , CODE(600),IDU(600),IELM
 0003              COMMON / ELM / S(18,18),LM(6),T,ETR,EPRM,ELGT,G, ST(3,18,4),INTL
 0004              DIMENSION  PHYX(3,6),PHYY(3,6),ALAM(12,18),           AL(3),AL2(3),
                  1  QMAT(6,6),AST(18,18),ZETA(3),AK21(3,18),DUM(6,6)
 0005              DIMENSION STA(14,14),STR(3,14,4),STRA(3,12,4),IK(6)
            C           INITIALIZE STIFFNESS MATRIX
 0006              DO 450 I = 1,14
 0007              DO 450 J = 1,14
 0008          450 STA(I,J) = 0.0
 0009              DO 475 I = 1,3
 0010              DO 475 J = 1,14
```

```
      DO 475 K = 1,4
  475 STR(I,J,K) = 0.0
      IJK = 1
      IF (IELM .EQ. 2) IJK = 2
      DO 500 JKL = 1,IJK
      DO 50  I=1,18
      DO 50  J=1,18
   50 S(I,J)=0.0
      IP = LM(1)
      IQ = LM(2)
      IR = LM(3)
      IF (JKL .EQ. 1) GO TO 80
      IP = LM(4)
      IQ = LM(3)
      IR = LM(2)
   80 CONTINUE
 6000 FORMAT ( 6I5 )
C                SET UP TRANSFORMATION INTO LOCAL COORDINATES
      XQP = X(IQ)-X(IP)
      YQP = Y(IQ)-Y(IP)
      ZQP = Z(IQ)-Z(IP)
      XRP = X(IR)-X(IP)
      YRP = Y(IR)-Y(IP)
      ZRP = Z(IR)-Z(IP)
      XRQ = X(IR)-X(IQ)
      YRQ = Y(IR)-Y(IQ)
      ZRQ = Z(IR)-Z(IQ)
      IF (JKL .EQ. 2) GO TO 170
      D1= SQRT(XQP*XQP+YQP*YQP+ZQP*ZQP)
 6100 FORMAT (6F 15.5)
      IF(D1)101,101,100
  100 AL(1)= XQP/D1
      AL(2)= YQP/D1
      AL(3)= ZQP/D1
      GO TO 102
  101 AL(1)=0.0
```


```
      AL(2)=0.0
      AL(3)=0.0
  102 RR= AL(1)*XRP +AL(2)*YRP +AL(3)*ZRP
      X2 = XRP -AL(1)*RR
      Y2 = YRP -AL(2)*RR
      Z2 = ZRP -AL(3)*RR
      D2 =SQRT(X2*X2 +Y2*Y2 +Z2*Z2)
      IF(D2)104,104,103
  103 AL2(1)=X2/D2
      AL2(2)=Y2/D2
      AL2(3)=Z2/D2
      GO TO 105
  104 AL2(1)=0.0
      AL2(2)=0.0
      AL2(3)=0.0
  105 CONTINUE
C     SET UP TRANSFORMATION MATRIX
      DO 221 I=1,12
      DO 221 J=1,18
  221 ALAM(I,J)=0.0
      KT=0
      NOD3 = 10
      IF (IELM .EQ. 2) NOD3 = 12
      IF (IMTL .GT. 3) NOD3 = 9
```

```
      DO 222 J=1,NOD3,2
      DO 222 I=1,3
      KT=KT+1
      ALAM(J,KT)= AL2(I)
  222 ALAM(J+1,KT)=AL(I)
C          TRANSFORM FROM GLOBAL TO LOCAL
  170 A1 =(XRQ*AL2(1)+YRQ*AL2(2)+ZRQ*AL2(3) )*(-1.)
      B1 = XRQ*AL  (1)+YRQ*AL  (2)+ZRQ*AL  (3)
      A2 =(XQP*AL2(1)+YQP*AL2(2)+ZQP*AL2(3))*(-1.)
      B2 = XQP*AL  (1)+YQP*AL  (2)+ZQP*AL  (3)
      A3 = XRP*AL2(1)+YRP*AL2(2)+ZRP*AL2(3)
      B3 =(XRP*AL  (1)+YRP*AL  (2)+ZRP*AL  (3)  )*(-1.)
      AREA2 = A2*B1 - A1*B2
      AI=1.0/AREA2
      IF (IMTL .GT. 3) GO TO 500
C     STIFFNESS AND STRESS MATRICES FOR 6 NODE TRIANGLE
C             SET UP PHYX
  180 CONTINUE
      PHYX(2,1) = -B1*AI
      PHYX(1,1) =-3.0*PHYX(2,1)
      PHYX(3,1) = PHYX(2,1)
      PHYX(1,2) = -B2*AI
      PHYX(2,2) = -3.0*PHYX(1,2)
      PHYX(3,2) = PHYX(1,2)
      PHYX(1,3) = -B3*AI
```



```
      PHYX(2,3) = PHYX(1,3)
      PHYX(3,3) = -3.0*PHYX(1,3)
      PHYX(1,4) = -4.0* PHYX(1,2)
      PHYX(2,4) = -4.0* PHYX(2,1)
      PHYX(3,4) = 0.0
      PHYX(1,5) = 0.0
      PHYX(2,5) = -4.0* PHYX(1,3)
      PHYX(3,5) = PHYX(1,4)
      PHYX(1,6) = PHYX(2,5)
      PHYX(2,6) = 0.0
      PHYX(3,6) = PHYX(2,4)
C             SET UP PHYY
      PHYY(2,1) = -A1*AI
      PHYY(1,1) = -3.0*PHYY(2,1)
      PHYY(3,1) = PHYY(2,1)
      PHYY(1,2) = -A2*AI
      PHYY(2,2) = -3.0*PHYY(1,2)
      PHYY(3,2) = PHYY(1,2)
      PHYY(1,3) = -A3*AI
      PHYY(2,3) = PHYY(1,3)
      PHYY(3,3) = -3.0*PHYY(1,3)
      PHYY(1,4) = -4.0*PHYY(1,2)
      PHYY(2,4) = -4.0*PHYY(2,1)
      PHYY(3,4) = 0.0
      PHYY(1,5) =0.0
      PHYY(2,5) = -4.0*PHYY(1,3)
      PHYY(3,5) = PHYY(1,4)
      PHYY(1,6) = PHYY(2,5)
      PHYY(2,6) = 0.0
      PHYY(3,6) = PHYY(2,4)
C         GENERATE  STIFFNESS MATRIX IN LOCAL CO-ORDS.
      DO 200 I=1,3
      DO 200 J=1,3
  200 QMAT(I,J) =      T*AREA2/24.
      DO 201 I=1,3
```

```
  201 QMAT(I,I)= 2.0*QMAT(I,I)
      CALL TRPRD (PHYX,QMAT,PHYX,DUM,ETR ,3,6)
      DO 202 I=1,6
      DO 202 J=1,6
  202 S(I,J) = S(I,J) + DUM(I,J)
      CALL TRPRD (PHYY,QMAT,PHYY,DUM,G    ,3,6)
      DO 205 I=1,6
      DO 205 J=1,6
  205 S(I,J) = S(I,J) + DUM(I,J)
      CALL TRPRD (PHYX,QMAT,PHYY,DUM,EPRM,3,6)
      DO 208 I=1,6
      DO 208 J=1,6
  208 S(I,J+6) = S(I,J+6) + DUM(I,J)
      CALL TRPRD (PHYY,QMAT,PHYX,DUM,G    ,3,6)
```



```
      DO 211 I=1,6
      DO 211 J=1,6
  211 S(I,J+6) = S(I,J+6) + DUM(I,J)
      CALL TRPRD (PHYY,QMAT,PHYY,DUM,ELGT,3,6)
      DO 214 I=1,6
      DO 214 J=1,6
  214 S(I+6,J+6) = S(I+6,J+6) + DUM(I,J)
      CALL TRPRD (PHYX,QMAT,PHYX,DUM,G    ,3,6)
      DO 217 I=1,6
      DO 217 J=1,6
  217 S(I+6,J+6)=S(I+6,J+6) + DUM(I,J)
C        COMPLETE LOCAL MATRIX USING SYMMETRY AND RE-ARRANGE ROWS & COLS
      KT= 0
      DO 218 I=2,12
      KT=KT+1
      DO 218 J=1,KT
  218 S(I,J) = S(J,I)
      DO 219 I=1,12
      AST(1,I) = S(1,I)
      AST(2,I) = S(7,I)
      AST(3,I) = S(2,I)
      AST(4,I) = S(8,I)
      AST(5,I) = S(3,I)
      AST(6,I) = S(9,I)
      AST(7,I) = S(4,I)
      AST(8 ,I)= S(10,I)
      AST(9 ,I)= S(5,I)
      AST(11,I)= S(6,I)
      AST(12,I)= S(12,I)
  219 AST(10,I)= S(11,I)
      DO 220 I=1,12
      S(I,1) = AST(I,1)
      S(I,2) = AST(I,7)
      S(I,3) = AST(I,2)
      S(I,4) = AST(I,8)
      S(I,5) = AST(I,3)
      S(I,6) = AST(I,9)
      S(I,7) = AST(I,4)
      S(I,8 )= AST(I,10)
      S(I,9 )= AST(I,5)
      S(I,11)= AST(I,6)
      S(I,12)= AST(I,12)
  220 S(I,10)= AST(I,11)
 6010 FORMAT  ( 12F10.0)
C       ELIMINATE ROWS AND COLS CORRESPONDING TO NODE 6
      DET = S(11,11)*S(12,12) - S(11,12)*S(12,11)
```

```
      IF (DET) 230,240,230
  230 AST(1,1) = S(12,12)/DET
      AST(2,2) = S(11,11)/DET
```



```
      AST(1,2) = S(11,12)/DET*(-1.)
      AST(2,1) = AST(1,2)
      DO 501 I = 1,2
      DO 501 J = 1,10
      AK21(I,J) = 0.0
      DO 501 K = 1,2
  501 AK21(I,J) = AK21(I,J) + AST(I,K)*S(K+10,J)
      DO 503 I=1,10
      DO 503 J=1,10
      AST(I,J)=0.0
      DO 503 K=1,2
  503 AST(I,J)=AST(I,J)+ S(I,K+10)*AK21(K,J)
      DO 504 I=1,10
      DO 504 J=1,10
  504 S(I,J) = S(I,J)- AST(I,J)
 6500 FORMAT ( 12F1C.0 )
C          SET UP APPROPRIATE  STRESS MATRICES
  240 ILK = 1
      IKL = 4
      IF ( IELM.EQ.1 ) GO TO 250
      IF ( JKL .EQ.1 ) IKL= 3
      IF ( JKL .EQ.2 ) ILK= 2
  250 DO 300 L = ILK,IKL
      DO 301 I=1,3
  301 ZETA(I)=0.0
      IF ( JKL.EQ.2 )  GO TO 302
      GO TO (303,304,307,308),L
  302 GO TO  ( 308,307,304,303 ) , L
  303 ZETA(1)=1.0
      GO TO 305
  304 ZETA(3)=1.0
      GO TO 305
  307 ZETA(2)=1.0
      GO TO 305
  308 DO 309 I=1,3
  309 ZETA(I)=1.0/3.0
  305 CONTINUE
      DO 310 I = 1,3
      DO 310 J = 1,18
  310 ST(I,J,L) = 0.0
      ST(1,1,L) = (4.*ZETA(1)-1.)*B1*AI
      ST(1,2,L) = (4.*ZETA(2)-1.)*B2*AI
      ST(1,3,L) = (4.*ZETA(3)-1.)*B3*AI
      ST(1,4,L) = 4.*(ZETA(2)*B1+ZETA(1)*B2)*AI
      ST(1,5,L) = 4.*(ZETA(3)*B2+ZETA(2)*B3)*AI
      ST(1,6,L) = 4.*(ZETA(1)*B3+ZETA(3)*B1)*AI
      ST(2,7,L) = (4.*ZETA(1)-1.)*A1*AI
      ST(2,8,L) = (4.*ZETA(2)-1.)*A2*AI
      ST(2,9,L) = (4.*ZETA(3)-1.)*A3*AI
      ST(2,10,L)= 4.*(ZETA(2)*A1+ZETA(1)*A2)*AI
      ST(2,11,L)= 4.*(ZETA(3)*A2+ZETA(2)*A3)*AI
      ST(2,12,L)= 4.*(ZETA(1)*A3+ZETA(3)*A1)*AI
      DO 410 I = 1,6
      ST(3,I,L)  = ST(2,I+6,L)
  410 ST(3,I+6,L)  = ST(1,I,L)
C           REARANGE STRESS MATRIX
      DO 416 I=1,3
```

```
0240              AST(I,1 ) = ST(I,1 ,L)
0241              AST(I,2 ) = ST(I,7 ,L)
0242              AST(I,3 ) = ST(I,2 ,L)
0243              AST(I,4 ) = ST(I,8 ,L)
0244              AST(I,5 ) = ST(I,3 ,L)
0245              AST(I,6 ) = ST(I,9 ,L)
0246              AST(I,7 ) = ST(I,4 ,L)
0247              AST(I,8 ) = ST(I,10,L)
0248              AST(I,9 ) = ST(I,5 ,L)
0249              AST(I,10) = ST(I,11,L)
0250              AST(I,11) = ST(I,6 ,L)
0251          416 AST(I,12) = ST(I,12,L)
            C     ELIMINATE NODE 6  FROM STRESS MATRIX
0252              DO 601 I=1,3
0253              DO 601 J=1,10
0254              ST(I,J,L) = 0.0
0255              DO 601 K=1,2
0256          601 ST(I,J,L) = ST(I,J,L) + AST(I,K+10)*AK21(K,J)
0257              DO 602 I=1,3
0258              DO 602 J=1,10
0259          602 AST(I,J) = AST(I,J) - ST(I,J,L)
0260              DO 605 I = 1,3
0261              DO 605 J = 1,10
0262          605 ST(I,J,L) = AST(I,J)
0263          300 CONTINUE
0264              IF (IELM .EQ. 1) GO TO 500
            C    ADD TWO TRIANGLES TO GET A SEVEN NODE QUADRALATERAL
0265              IF (JKL .EQ. 2) GO TO 460
0266              IK(1) = 1
0267              IK(2) = 3
0268              IK(3) = 2
0269              IK(4) = 5
0270              IK(5) = 7
0271              GO TO 470
0272          460 IK(1) = 4
0273              IK(2) = 2
0274              IK(3) = 3
0275              IK(4) = 6
0276              IK(5) = 7
0277          470 DO 480 K = ILK,IKL
0278              DO 480 I = 1,5
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0007

```
0279              K3 = IK(I)*2 - 2
0280              DO 480 L = 1,2
0281              DO 430 J = 1,3
0282          480 STR(J,K3+L,K) = STR(J,K3+L,K) + ST(J,2*I-2+L,K)
0283              DO 490 I = 1,5
0284              DO 490 J = 1,5
0285              K3 = IK(I)*2 - 2
0286              K4 = IK(J)*2 - 2
0287              DO 490 L = 1,2
0288              DO 490 M = 1,2
0289          490 STA(K3+L,K4+M) = STA(K3+L,K4+M) + S(2*I-2+L,2*J-2+M)
0290          500 CONTINUE
0291              IF (IMTL .GT. 3) GO TO 700
0292              IF (IELM .EQ. 1) GO TO 560
0293              DO 495 I = 1,14
0294          495 WRITE (6,6500) (STA(I,J),J=1,14)
            C     REMOVE NODE SEVEN FROM QUAD
0295              DET = STA(13,13)*STA(14,14) - STA(13,14)*STA(13,14)
0296              IF (ABS(DET) .LT. 1.0 E-10) WRITE (6,6200)
0297         6200 FORMAT ( /// ,5X, 26HREDUCED MATRIX IS SINGULAR  )
```

```
      IF (ABS(DET) .LT. 1.0 E-10) DET = 1.0
      QMAT(1,1) = STA(14,14)/DET
      QMAT(2,2) = STA(13,13)/DET
      QMAT (1,2) = STA(13,14)/DET*(-1.)
      QMAT(2,1) = QMAT(1,2)
      DO 510 I = 1,2
      DO 510 J = 1,12
      AK21(I,J) = 0.0
      DO 510 IJ = 1,2
  510 AK21(I,J) = AK21(I,J) + QMAT(I,IJ)*STA(IJ+12,J)
      DO 520 I = 1,12
      DO 520 J = 1,12
      AST(I,J) = 0.0
      DO 520 IJ = 1,2
  520 AST(I,J) = AST(I,J) + STA(I,IJ+12)*AK21(IJ,J)
      DO 530 I = 1,12
      DO 530 J = 1,12
  530 S(I,J) = STA(I,J) - AST(I,J)
      DO 550 K = 1,4
      DO 540 I = 1,3
      DO 540 J = 1,12
      STRA(I,J,K) = 0.0
      DO 540 IJ = 1,2
  540 STRA(I,J,K) = STRA(I,J,K) + STR(I,IJ+12,K)*AK21(IJ,J)
      DO 550 I = 1,3
      DO 550 J = 1,12
  550 ST(I,J,K) = STR(I,J,K) - STRA(I,J,K)
      DO 555 K = 2,3
      DO 555 I = 1,3
```



```
      DO 555 J = 1,12
  555 ST(I,J,K) = ST(I,J,K)/2.0
      DO 556 I = 1,12
  556 WRITE (6,6500) (S(I,J),J=1,12)
C      GENERATE STIFFNESS MATRIX IN GLOBAL CO-ORDINATES
  560 CALL TRPRD (ALAM,S,ALAM,S,1.0,12,18)
C         TRANSFORM STRESS MATRIX INTO GLOBAL CO-ORDINATES
      DO 417 L = 1,4
      DO 415 I = 1,3
      DO 415 J = 1,12
  415 AST(I,J) = ST(I,J,L)
      DO 417 I=1,3
      DO 417 J=1,18
      ST(I,J,L) = 0.0
      DO 417 K=1,12
  417 ST(I,J,L) = ST(I,J,L) + AST(I,K)*ALAM(K,J)
      RETURN
C     STIFFNESS AND STRESS MATRICES FOR 3 NODE TRIANGLE
  700 CONTINUE
      DO 750 I = 1,3
      DO 750 J = 1,6
      QMAT(I,J) = 0.0
  750 PHYX(I,J) = 0.0
      PHYX(1,1) = B1
      PHYX(1,3) = B3
      PHYX(1,5) = B2
      PHYX(2,2) = A1
      PHYX(2,4) = A3
      PHYX(2,6) = A2
      PHYX(3,1) = A1
      PHYX(3,2) = B1
```

```
      PHYX(3,3) = A3
      PHYX(3,4) = B3
      PHYX(3,5) = A2
      PHYX(3,6) = B2
      DO 760 I = 1,3
      DO 760 J = 1,6
      QMAT(I,J) = 0.0
      PHYX(I,J) = PHYX(I,J)/AREA2
  760 CONTINUE
      DO 765 I = 1,3
      DO 765 J = 1,9
      ST(I,J,1) = 0.0
      DO 765 K = 1,6
  765 ST(I,J,1) = ST(I,J,1) + PHYX(I,K)*ALAM(K,J)
      QMAT(1,1) = ETR
      QMAT(2,2) = ELGT
      QMAT(1,2) = EPRM
      QMAT(2,1) = EPRM
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0009

```
      QMAT(3,3) = G
      W = T*AREA2/2.
      CALL TRPRD (PHYX,QMAT,PHYX,DUM,W,3,6)
      DO 780 I = 1,9
      DO 780 J = 1,6
      AST(I,J) = 0.0
      DO 780 K = 1,6
  780 AST(I,J) = AST(I,J) + ALAM(K,I)*DUM(K,J)
      DO 790 I = 1,9
      DO 790 J = 1,9
      S(I,J) = 0.0
      DO 790 K = 1,6
  790 S(I,J) = S(I,J) + AST(I,K)*ALAM(K,J)
      RETURN
      END
```

COMMON BLOCK / XYZ / MAP SIZE 003CF4

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
X	000000	Y	000480	Z	000960	UX	000E10	UY	001770
UZ	0020D0	CODE	002A30	IDO	003390	IELM	003CF0		

COMMON BLOCK / ELM / MAP SIZE 0008A0

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
S	000000	LM	000510	T	000528	ETR	00052C	EPRM	000530
ELGT	000534	G	000538	ST	00053C	INTL	00089C		

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
I	000328	J	00032C	K	000330	IJK	000334	JKL	000338
IP	00033C	IQ	000340	IR	000344	XQP	000348	YQP	00034C
ZQP	000350	XRP	000354	YRP	000358	ZRP	00035C	XRQ	000360
YRQ	000364	ZRQ	000368	D1	00036C	RR	000370	X2	000374
Y2	000378	Z2	00037C	D2	000380	KT	000384	NOD3	000388
A1	00038C	B1	000390	A2	000394	B2	000398	A3	00039C
B3	0003A0	AREA2	0003A4	AI	0003A8	DET	0003AC	ILK	0003B0
IKL	0003B4	L	0003B8	K3	00038C	K4	0003C0	M	0003C4
IJ	0003C8	W	0003CC						

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
PHYX	0003D0	PHYY	000418	ALAM	000460	AL	0007C0	AL2	0007CC
QMAT	0007D8	AST	000868	ZETA	000D78	AK21	000D84	DUM	000E5C
STA	000EEC	STR	0011FC	STRA	00149C	IK	00160C		

```
                                        SUBPROGRAMS CALLED
SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
TRPRD      0016F4        IBCOM#     0016F8        SORT       0016FC

                                           LABEL MAP
LABEL      LOCATION      LABEL      LOCATION      LABEL      LOCATION      LABEL      LOCATION      LABEL      LOCATION
  450      00184A          475      0018AC           50      001958           80      0019E0         6000      0019EC
 6100      001AC6          100      001AE0          101      00180A          102      001826          103      0018BE
  104      001BE8          105      001C04          221      001C1C          222      001CF0          170      001D1C
  180      001E28          200      001FC8          201      00201E          202      002074          205      0020F6
  208      002178          211      0021F6          214      002274          217      0022F2          218      002376
  219      00244C          220      0024DA         6010      0024F4          230      002526          501      00259A
  503      00265C          504      0026F4         6500      002742          240      002750          250      0027A6
  301      00278A          302      002824          303      002858          304      00286A          307      00287C
  308      00288E          309      00289A          305      0028C4          310      0028DC          410      002AA2
  416      002B3E          601      002BA4          602      002C46          605      002CC0          300      002D22
  460      002D86          470      002D86          480      002DFE          490      002F4A          500      003032
  495      003074         6200      003134          510      0031F6          520      003280          530      003348
  540      0033FC          550      0034AC          555      003572          556      0035E0          560      00365A

FORTRAN IV     MODEL 44    PS          VERSION 3,  LEVEL 4    DATE  71321                            PAGE  0011

  415      00368C          417      003736          700      0037DE          750      003814          760      003902
  765      003964          780      003A68          790      003B2A

 TOTAL MEMORY REQUIREMENTS 003C78 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//TRPRD  EXEC FORTRAN(BCD,MAP)

                 SUBROUTINE TRPRD (A,B,C,E,FACT,N1,N2)
                 DIMENSION A(N1,N2),B(N2,N2),C(N1,N2),D(18,18),E(N2,N2)
           C       A SUBROUTINE  TO PERFORM   A-TRANS * B * C  MULTIPLICATION
           C       RETURN SOLUTION IN E
                 DO 100 I= 1,N2
                 DO 100 J= 1,N1
                 D(I,J)=0.0
                 DO 100 K= 1,N1
             100 D(I,J) = D(I,J) + A(K,I)  *B(K,J)
                 DO 200 I=1,N2
                 DO 200 J=1,N2
                 E(I,J) = 0.0
                 DO 200 K=1,N1
             200 E(I,J) = E(I,J) + D(I,K)*C(K,J)*FACT
                 RETURN
                 END

FORTRAN IV     MODEL 44    PS          VERSION 3,  LEVEL 4    DATE  71321                            PAGE  0002

                                           SCALAR MAP
SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
N1         0000E8        N2         0000EC        I          0000F0        J          0000F4        K          0000F8
FACT       0000FC

                                           ARRAY MAP
SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION      SYMBOL     LOCATION
A          000100        B          000104        C          000108        D          00010C        E          00061C
```

```
                                              LABEL MAP
LABEL      LOCATION       LABEL     LOCATION       LABEL      LOCATION       LABEL      LOCATION       LABEL      LOCATION
  100      000798           200     00087A

  TOTAL MEMORY REQUIREMENTS 0009A4 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//SOLPAC EXEC FORTRAN(BCD,MAP)

FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                         PAGE  0001

      C     ************************ OVERLAY C ****************************************
            SUBROUTINE SOLPAC
            COMMON / COMN / NUMNP,NUMEL,NLOD,NLIN,MBAND,NUMBLK,NBN,NEL(20)
            COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLD,NR1,NR2
            COMMON /DUMP / ITEST
            DIMENSION DIAG(1800),AI(2700),AJ(2700), B(1800),X(1800)
            EQUIVALENCE  ( X(1),B(1) ,AJ(1))
            REAL*8  DSUM,DA,DB
            NRS=3*NUMNP
            NR4=NR4DIA
            NRD=1
            CALL OUTIN (2,DIAG  ,NRS  )
            DIAG(NRD)= SQRT(DIAG(NRD))
            NRD=1
            M=MBAND-1
            NE=3*M
            DO 1010 II=1,NUMNP
            NR4=(II-1)*NRECST+NR4STF
            LR4=NR4
            IFRED=0
            IL=3*(II-1)+1
            IF ( IL.EQ.1 ) IFRED=1
            IF(IL.EQ.1) IL=2
            IR=3*II
            CALL OUTIN (2,AI    ,NE   )
            DO 100 I=IL,IR
            IFRED=IFRED+1
            IMM=I-M
             IF(IMM)101,101,102
        101 IMM=1
        102 IM1=I-1
            N1=MBAND-I+(IFRED-1)*M
            INITAL=0
            DO 200  J=IMM,IM1
            JFRED =MOD(J-1,3) +1
            N2=MBAND-J+ (JFRED-1)*M
            NRD=J
            DSUM=0.0
             JM1=J-1
            IF ( INITAL .GT. 0 )  GO TO 150
            JJ= (J+2) /3
            NR4= (JJ-1)  *NRECST+NR4STF
            CALL OUTIN (2,AJ    ,NE   )
            INITAL=1
        150 CONTINUE
            IF ( JFRED    .EQ. 3 )  INITAL = 0
            IF ( II.NE.JJ.OR.JFRED.GE.IFRED )  GO TO 170
            JL = (JFRED-1)*M + 1
            JR = JL + M-1
```

```
      DO 160 L = JL,JR
      AJ(L) = AI(L)
  160 CONTINUE
  170 CONTINUE
      IF(IMM-JM1)201,201,202
  201 DO 300 K=IMM,JM1
      K1=K+N1
      K2 =K+N2
       DA=DBLE(AI(K1))
      DB=DBLE(AJ(K2))
       DSUM=DSUM+DA*DB
  300 CONTINUE
  202 CONTINUE
       J1=J+N1
  200 AI(J1)=(AI(J1)-SNGL(DSUM))/DIAG(NRD)
      NRD=I
      DSUM=0.0
       DO 400 K=IMM,IM1
       K1=N1+K
      DA=DBLE(AI(K1))
  400 DSUM=DSUM+DA*DA
      NRD=I
C***********TEST POS.  DEF. AND SINGULARITY****************
      IF(DIAG(NRD)-SNGL(DSUM))204,205,206
  204 NERR=-1
      GO TO 207
  205 NERR=0
207   WRITE(6,203) NRD,NERR,DIAG(NRD),DSUM
203   FORMAT(1H0,4HNRD=,I5,5X,5HNERR=,I5,2E15.5)
      DIAG(NRD)= DABS(DSUM) +1.E+12
  206 NERR=1
      DIAG(NRD)=SQRT(DIAG(NRD)-SNGL(DSUM))
  100 CONTINUE
      NR4=LR4
      CALL OUTIN (1,AI    ,NE   )
 1010 CONTINUE
C*********BEGIN FORWARD SUBSTITUTION************************
      DO 500 K=1,NLOD
      NR4= (K-1)*NRECLD+NR4LOD
      LR4= NR4
      NRD=1
      CALL OUTIN (2,B      ,NRS  )
         X(1)=B(1)/DIAG(NRD)
      DO 6010 JJ=1,NUMNP
      IL = 3*(JJ-1)+1
      JFRED=0
      IF ( IL.EQ.1 ) JFRED=1
      IF (IL.EQ.1) IL = 2
      IR = 3*JJ
      NR4=(JJ-1) *NRECST+1
      CALL OUTIN (2,AI    ,NE   )
      DO 600 J=IL,IR
      JFRED=JFRED+1
      N1=MBAND-J+ (JFRED-1)*M
      NRD=J
      JMM=J-M
      IF(JMM)601,601,602
  601  JMM=1
  602  JM1=J-1
      DSUM=0.0
      DO 603 I=JMM,JM1
```

```
            DA=DBLE(AI(I+N1))
            DB=DBLE(X(I))
            DSUM=DSUM+DA*DB
      603 CONTINUE
            X(J)=B(J)-SNGL(DSUM)
            X(J)=X(J)/DIAG(NRD)
      600 CONTINUE
     6010 CONTINUE
            NR4=LR4
            CALL OUTIN (1,X      ,NRS  )
      500 CONTINUE
C************BEGIN BACK SUBSTITUTION************************
            DO 700 J=1,NLOD
            NR4 = (J-1)*NRECLD+NR4LOD
            LR4= NR4
            NRD=NRS
            CALL OUTIN (2,X      ,NRS  )
            B(NRS)=X(NRS)/DIAG(NRD)
            NM1=NRS-1
            DO 800 L=1,NM1
            I=NRS-L
            NRD=I
            IP1=I+1
            DSUM=0.0
            IF(NRS-IP1)802,803,803
      803 CONTINUE
            NUMNP1=IP1/3
            IF(3*NUMNP1.LT.IP1) NUMNP1=NUMNP1+1
            DO 801 LL=NUMNP1,NUMNP
            IL = 3*(LL-1)+1
            IR = 3*LL
            IF ( LL.EQ.NUMNP1 ) IL=IP1
            JFRED=MOD(IL-1,3)
            NR4=(LL-1)*NRECST+1
            CALL OUTIN (2,AI     ,NE   )
            DO 804 K=IL,IR
            JFRED=JFRED + 1
FORTRAN IV   MODEL 44  PS        VERSION 3, LEVEL 4   DATE 71321                PAGE 0004
            N1 = I + (JFRED-1)*M
            IF(I-(K-MBAND))802,802,805
      805 CONTINUE
            DB=DBLE(B(K))
            DA=DBLE(AI(N1+MBAND-K))
      804 DSUM=DSUM+DA*DB
      801 CONTINUE
      802 CONTINUE
            B(I)=X(I)-SNGL(DSUM)
            B(I)=B(I)/DIAG(NRD)
      800 CONTINUE
            NR4= LR4
            CALL OUTIN (1,B      ,NRS  )
      700 CONTINUE
            NERR=1
            RETURN
            END
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                         PAGE  0005

                                        COMMON BLOCK / COMN    / MAP SIZE  00006C
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
NUMNP     000000      NUMEL     000004      NLOD      000008      NLIN      00000C      MBAND     000010
NUMBLK    000014      NBN       000018      NEL       00001C

                                        COMMON BLOCK / RAX     / MAP SIZE  000024
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
NR4       000000      NR4LOD    000004      NR4STF    000008      NR4DIA    00000C      NREC      000010
NRECST    000014      NRECLD    000018      NR1       00001C      NR2       000020

                                        COMMON BLOCK / DUMP    / MAP SIZE  000004
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
ITEST     000000

                                        EQUIVALENCE DATA MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
X         000184      B         000184      AJ        000184

                                        SCALAR MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
NRS       0028B4      NRD       0028B8      M         0028BC      NE        0028C0      II        0028C4
LR4       0028C8      IFRED     0028CC      IL        0028D0      IR        0028D4      I         0028D8
IMM       0028DC      IM1       0028E0      N1        0028E4      INITAL    0028E8      J         0028EC
JFRED     0028F0      N2        0028F4      JM1       0028F8      JJ        0028FC      JL        002C00
JR        002C04      L         002C08      K         002C0C      K1        002C10      K2        002C14
J1        002C18      MERR      002C1C      JMM       002C20      NM1       002C24      IP1       002C28
NUMNP1    002C2C      LL        002C30      DSUM      002C38      DA        002C40      DB        002C48

                                        ARRAY MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
DIAG      002C50      AI        004870

                                        SUBPROGRAMS CALLED
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
OUTIN     007240      IBCOM#    0072A4      SQRT      0072A8

                                        LABEL MAP
LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION
   101    007510         102    007520         150    007634         160    0076DA         170    0076EC
   201    007700         300    007768         202    00777A         200    00778A         400    00780E
   204    007856         205    00786C         207    00787C         203    0078C4         206    0078FC
   100    007932        1010    00795A         601    007ADA         602    007AEA         603    007B58
   600    007B92        6010    007BA4         500    007BD8         803    007CBA         805    007DE2
   804    007E28         801    007E4A         802    007E60         800    007E8C         700    007EBC

 TOTAL MEMORY REQUIREMENTS 007F60 BYTES

FORTRAN IV    MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                         PAGE  0006

COMPILER HIGHEST SEVERITY CODE WAS  0
//DEFL    EXEC FORTRAN(BCD,MAP)
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3,  LEVEL 4   DATE  71321                          PAGE  0001

               C      ************************* OVERLAY D ******************************************
 0001                 SUBROUTINE DEFL
               C
 0002                 COMMON / COMN / NUMNP,NUMEL,NLOD,MLIN,MBAND,NUMBLK,NBN,NEL(20)
 0003                 COMMON / MATL / E(4,5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
 0004                 COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
                     1 ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
 0005                 COMMON / LODA / XLOAD,YLOAD,XYLOD,XMOM,YMOM,XYMOM,TORQ,XQSHR,YQSHR
 0006                 COMMON /LODB / APPLD(9,10)
 0007                 COMMON/RAX/NR4,NR4LOD,NR4STF,NR4DIA,NREC,NRECST,NRECLD,NR1,NR2
 0008                 DIMENSION P(18),LM(6),STRESS(4,3),FBKL(5)
                     1  ,ST(3,18,4),MM(6),STRAIN(4,3)
 0009                 DIMENSION IDO(300),B( 900)
               C
               C      SOLVE FOR DISPLACEMENTS
 0010                 BACKSPACE 3
 0011                 NODES = NUMNP
 0012                 READ (3)  (IDO(I),I=1,NODES)                                    TAPE 3
 0013                 CALL BUKL (FBKL)
 0014                 NRS=3*NUMNP
 0015                 NR4=NR4LOD
 0016                 DO 320 IZ = 1,NLOD
 0017                 CALL OUTIN (2,B      ,NRS   )
 0018                 IF (ICNT .GT. 1 .AND. IOPT .EQ. 1) GO TO 102
 0019                 MPRINT=0
 0020                 DO 100 N = 1,NODES
 0021                 IF(MPRINT) 90,80,90
 0022              80 WRITE(6,6100) IZ
 0023            6100 FORMAT (1H1, 20X, 11HDEFLECTIONS, 10X, 9HLOAD CASE, I4, //1H
                     1 ,6HNODAL / 6H POINT, 14X, 2HUX, 22X,2HUY,  22X, 2HUZ, // )
 0024                 MPRINT=50
 0025              90 MPRINT=MPRINT-1
 0026             100 WRITE (6,6110)IDO(N),B(3*N-2),B(3*N-1),B(3*N)
 0027            6110 FORMAT (I17,  3E24.7)
               C      COMPUTE ELEMENT STRESSES
 0028             102  MPRINT=0
 0029                 REWIND 3                                                         TAPE 3
 0030                 DO 110 I = 1,10
 0031             110 CMS(I) = 10.0
 0032                 XMS = 10.0
 0033                 DO 310 IJ= 1,NUMEL
 0034                 READ (3) IOP,ST,KPO  ,KQO     ,KRO,KSO ,LM,E22,E11,E12,E33       TAPE3
                     1                     ,TO      ,INTL              ,MM,NNOD
 0035                 DO 120 I=1,NNOD
 0036                 II=3*I
 0037                  JJ=3*MM(I)
 0038                  P(II-2)=B(JJ-2)
 0039                  P(II-1)=B(JJ-1)
 0040             120 P(II)=B(JJ)

               C*********************COMPUTE STRESSES AT FOUR LOCATIONS************************
 0041                 KJ = 4
 0042                 IF (NNOD .EQ. 3) KJ = 1
 0043                 NOD3 = 3*NNOD
 0044                  DO 122 I=1,KJ
 0045                 DO 123 J=1,3
 0046                 STRAIN(I,J) = 0.0
 0047                  DO 123 K=1,NOD3
 0048             123 STRAIN(I,J) = STRAIN(I,J) + ST(J,K,I)*P(K)
 0049                 STRESS(I,J) = STRAIN(I,1)*E22       + STRAIN(I,2)*E12
```

```
      STRESS(I,2) = STRAIN(I,1)*E12       + STRAIN(I,2)*E11
      STRESS(I,3) = STRAIN(I, 3)*E33
  122 CONTINUE
      CALL MARGIN  (INTL,STRESS,XMS,FBKL)
      IF (NLIN .GT. 0) CALL MODUL (INTL,IDP,STRAIN,E11     ,E22
     1  ,E12     ,E33       )
      IF (ICNT .GT. 1 .AND . IOPT .EQ. 1) GO TO 160
      IF(MPRINT) 150,140,150
  140 WRITE (6,6120)
 6120 FORMAT ( 1H1, 9H  ELEMENT , /, 21H   NO   I   J   K   L ,12X
     1  , 7HSIGMA 1, 10X, 7HSIGMA 2, 12X, 6HTAU 12, 18X, 6HMARGIN
     1  , 10X, 8H  LEG      // )
      MPRINT = 10
  150 MPRINT=MPRINT-1
      WRITE(6,6130) IDP,KPO     ,KQO     ,KRO,KSO , STRESS(1,2)
     1  ,STRESS(1,1),STRESS(1,3),XMS,INTL
 6130 FORMAT ( / ,1X, I5,4I4,4F19.3,I15)
      IF (NNOD-3) 160,160,152
  152 DO 155 I=2,4
  155 WRITE(6,6140)STRESS(I,2),STRESS(I,1),STRESS(I,3)
 6140 FORMAT(22X,3F19.3)
  160 CONTINUE
  300 CONTINUE
      NR2=IJ
      WRITE(2'NR2)
     *          I,KPO   ,KQO   ,KRO,KSD,E22   ,E11   ,E12   ,E33         TAPE 2
     1  ,TO   ,INTL
  310 CONTINUE
      IF (ICNT .GT. 1 .AND . IOPT .EQ. 1) GO TO 320
      N = 3
      IF (IFACE .EQ. 1) N = 5
      WRITE (6,6150) (CMS(I),I=1,N)
 6150 FORMAT ( 1H1, 5X, 30HMARGIN OF SAFETY IN EACH LEG # , 5F10.5 )
      XLOAD = APPLD(1,IZ)
      YLOAD = APPLD(2,IZ)
      XYLOD = APPLD(3,IZ)
      XMOM  = APPLD(4,IZ)
      YMOM  = APPLD(5,IZ)
      XYMOM = APPLD(6,IZ)
FORTRAN IV    MODEL 44  PS       VERSION 3,  LEVEL 4   DATE  71321                    PAGE  0003

      TORQ  = APPLD(7,IZ)
      XQSHR = APPLD(8,IZ)
      YQSHR = APPLD(9,IZ)
      YMS = XMS
      IF (YMS .GT. .95) YMS = .95
      XALOW = XLOAD/(1.0-YMS)**.5
      YALOW = YLOAD/(1.0-YMS)**.5
      XYALW = XYLOD/(1.0-YMS)**.5
      XMALW = XMOM /(1.0-YMS)**.5
      YMALW = YMOM /(1.0-YMS)**.5
      XYMAL = XYMOM/(1.0-YMS)**.5
      TROAL = TORQ /(1.0-YMS) **.5
      XQSAL = XQSHR/(1.0-YMS)**.5
      YQSAL = YQSHR/(1.0-YMS)**.5
      WRITE (6,6160) XALOW,YALOW,XYALW,XMALW,YMALW,XYMAL,TROAL
     1  ,XQSAL,YQSAL
 6160 FORMAT ( /// ,20X,14HULTIMATE LOADS,/, 9X,2HNX, 10X, 2HNY, 9X,
     1 3HN/Y, 10X, 2HMX, 10X,2HMY, 10X, 3HMXY, 6X, 6HTORQUE, 8X,
     1  2HQX, 10X, 2HQY, / , 9F12.0 )
      Y1LGT = .28867 + Y1OFF
      Y2LGT = .86657 + Y2OFF
```

```
      XLGT = ((NX-1)*Y2LGT + Y1LGT)*SIDE
      YLGT = (FLOAT(NY)-.5)*SIDE
C     CALCULATE STIFFNESSES
      IF (ABS(XLOAD) .LT. .001) GO TO 340
      STRS = XLOAD/(HT*XLGT)
      IF(ITET .EQ. 1) GO TO 330
      J1 = (NX-1)
      J2 = (NX*NY*2 - NX - 1)
      YLT = YLGT
      GO TO 335
  330 J1 = (NY/2*(4*NY-1)+3)
      J2 = (NY/2+1)*(4*NY-1)
      YLT = SIDE*(NY-1)
  335 DO 336 K = 1,NUMNP
      IF (IDO(K) .EQ. J1) N1 = K*3 - 2
  336 IF (IDO(K) .EQ. J2) N2 = K*3 - 2
      STRN = (-B(N1)+B(N2))/YLT
      XMOD = STRS/STRN
      WRITE (6,6200) XMOD
 6200 FORMAT ( ///, 5X, 26H  MODULUS IN X DIRECTION #, F15.5, 3HPSI )
  340 IF (ABS(YLOAD) .LT. .001) GO TO 360
      STRS = YLOAD/(HT*YLGT)
      IF(ITET .EQ. 1) GO TO 350
      J1 = (NX*(NY-1)+1)
      J2 = NX*NY
      GO TO 355
  350 J1 = (2*NY-1)
      J2 = ((4*NY-1)*(NY-1)+2*NY-2)
  355 DO 356 K = 1,NUMNP
      IF (IDO(K) .EQ. J1) N1 = K*3 - 1
  356 IF (IDO(K) .EQ. J2) N2 = K*3 - 1
      STRN = (-B(N1)+B(N2)) /XLGT
      YMOD = STRS/STRN
      WRITE (6,6250) YMOD
 6250 FORMAT ( ///, 5X, 26H  MODULUS IN Y DIRECTION #, F15.5, 3H
  360 CONTINUE
  320 CONTINUE
      RETURN
      END
```

FORTRAN IV   MODEL 44 PS   VERSION 3, LEVEL 4   DATE 71321   PAGE 0005

COMMON BLOCK / COMN / MAP SIZE 00006C

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
NUMNP	000000	NUMEL	000004	NLOD	000008	NLIN	00000C	MBAND	000010
NUMBLK	000014	NBN	000018	NEL	00001C				

COMMON BLOCK / MATL / MAP SIZE 0000DC

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
E	000000	FXT	000050	FYT	000064	FXC	000078	FYC	00008C
SSS	0000A0	CMS	000084						

COMMON BLOCK / ONE / MAP SIZE 0000E8

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SIDE	000000	HT	000004	THK	000008	THETA	00001C	Y1OFF	000020
Y2OFF	000024	THT	000028	DUM	00002C	TC	000030	XC	00005C
MED	000088	NX	0000B8	NY	0000BC	IOPT	0000C0	ITYP	0000C4
IFACE	0000C8	IHOLE	0000CC	ITET	0000D0	ICNT	0000D4	NNEW	0000D8

```
                                         COMMON BLOCK / LODA    / MAP SIZE  000024
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
XLOAD     000000      YLOAD     000004      XYLOD     000008      XMOM      00000C      YMOM      000010
XYMOM     000014      TORQ      000018      XQSHR     00001C      YQSHR     000020

                                         COMMON BLOCK / LODB    / MAP SIZE  000168
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
APPLD     000000

                                         COMMON BLOCK / RAX     / MAP SIZE  000024
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
NR4       000000      NR4LOD    000004      NR4STF    000008      NR4DIA    00000C      NREC      000010
NRECST    000014      NRECLD    000018      NR1       00001C      NR2       000020

                                         SCALAR MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
NODES     0001B8      I         0001BC      NRS       0001C0      IZ        0001C4      MPRINT    0001C8
N         0001CC      XMS       0001D0      IJ        0001D4      IDP       0001D8      KPO       0001DC
KQO       0001F0      KRO       0001E4      KSO       0001E8      E22       0001EC      E11       0001F0
E12       0001F4      E33       0001F8      TO        0001FC      IMTL      000200      NNOD      000204
II        000208      JJ        00020C      KJ        000210      NOD3      000214      J         000218
K         00021C      YMS       000220      XALOW     000224      YALOW     000228      XYALW     00022C
XMALW     000230      YMALW     000234      XYMAL     000238      TROAL     00023C      XQSAL     000240
YQSAL     000244      YILGT     000248      Y2LGT     00024C      XLGT      000250      YLGT      000254
STRS      000258      J1        00025C      J2        000260      YLT       000264      N1        000268
N2        00026C      STRN      000270      XMOD      000274      YMOD      000278

                                         ARRAY MAP
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
P         00027C      LM        0002C4      STRESS    0002DC      FBKL      00030C      ST        000320
 FORTRAN IV    MODEL 44   PS          VERSION 3,  LEVEL 4   DATE  71321                          PAGE  0006

MM        000680      STRAIN    000698      100       0006C8      B         000878

                                         SUBPROGRAMS CALLED
SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION    SYMBOL    LOCATION
IBCOM#    001988      BUKL      00198C      OUTIN     001990      MARGIN    001994      MODUL     001998
FRXPR#    00199C

                                         LABEL MAP
LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION    LABEL     LOCATION
     80   00186E       6100     001894         90     0018F6        100    001C06       6110     001C6A
    102   001C7C        110     001CA0        120     001DC0        123    001E46        122     001EF4
    140   001F76       6120     001F94        150     002006       6130    002080        152     0020AA
    155   002086       6140     002106        160     002118        300    002118        310     002194
   6150   002254       6160     002490        330     0025CC        335    002640        336     002676
   6200   002708        340     00273A        350     00279E        355    0027E6        356     00281C
   6250   002880        360     0028E2        320     0028E2

  TOTAL MEMORY REQUIREMENTS 002974 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//MARGIN EXEC FORTRAN(BCD,MAP)
 FORTRAN IV    MODEL 44   PS          VERSION 3,  LEVEL 4   DATE  71321                          PAGE  0001

  0001              SUBROUTINE MARGIN (IMTL,STRESS,XMS,FBKL)
  0002              COMMON / MATL / E(4,5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
  0003              DIMENSION LM(6),FBKL(5),STRESS(4,3)
  0004              JK = 4
  0005              IF (IMTL .GT. 3) JK = 1
  0006              DO 190 J = 1,JK
```

```
      IF (STRESS(J,2)) 110,120,120
  110 RX = -STRESS(J,2)/FXC(IMTL)
      F1 = FXC(IMTL)
      GO TO 130
  120 RX =  STRESS(J,2)/FXT(IMTL)
      F1 = FXT(IMTL)
  130 CONTINUE
      IF (STRESS(J,1)) 140,150,150
  140 RY = -STRESS(J,1)/FYC(IMTL)
      F2 = FYC(IMTL)
      GO TO 160
  150 RY =  STRESS(J,1)/FYT(IMTL)
      F2 = FYT(IMTL)
  160 CONTINUE
      RS = ABS(STRESS(J,3))/SSS(IMTL)
      R = RX*RX - F2/F1*RX*RY + RY*RY + RS*RS
      IF (R .LT. 1.0 E-6) R = 1.0 E-6
      AMS = 1./SQRT(R) - 1.
      IF (    STRESS(J,2) ) 170,180,170
  170 BMS =-FBKL(IMTL)/STRESS(J,2) - 1.0
      IF (STRESS(J,2) .GE. 0.0) BMS = 10.
      IF (BMS .LT. AMS) AMS = BMS
  180 IF (AMS .LT. CMS(IMTL)) CMS(IMTL) = AMS
      IF (AMS .LT. XMS) XMS = AMS
  190 CONTINUE
  200 CONTINUE
      RETURN
      END
```

```
FORTRAN IV   MODEL 44  PS      VERSION 3, LEVEL 4   DATE  71321                    PAGE  0002
```

COMMON BLOCK / MATL / MAP SIZE 0000DC

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
E	000000	FXT	000050	FYT	000064	FXC	000078	FYC	00008C
SSS	0000A0	CMS	000084						

SCALAR MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
JK	000110	IMTL	000114	J	000118	RX	00011C	F1	000120
RY	000124	F2	000128	RS	00012C	R	000130	AMS	000134
BMS	000138	XMS	00013C						

ARRAY MAP

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
LM	000140	FBKL	000158	STRESS	00015C				

SUBPROGRAMS CALLED

SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION	SYMBOL	LOCATION
SQRT	000160								

LABEL MAP

LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION	LABEL	LOCATION
110	00024C	120	000280	130	0002AC	140	0002CC	150	000308
160	00033C	170	0003E2	180	000440	190	000480	200	000496

```
TOTAL MEMORY REQUIREMENTS 0004E4 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//MODUL  EXEC FORTRAN(BCD,MAP)
```

```
FORTRAN IV    MODEL 44  PS        VERSION 3, LEVEL 4   DATE  71321                              PAGE 0002

  0048              A4 = E2*NU12/XNU
  0049          210 CONTINUE
  0050              RETURN
  0051              END
                                       COMMON BLOCK / MODU   / MAP SIZE  000550
 SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
 SIGT1     000000        SIGT2     000084        SIGC1     000168        SIGC2     00021C        SIGSS     0002D0
 PRXT      000384        PRXC      000438        STT1      0004EC        STT2      000500        STC1      000514
 STC2      000528        STSS      00053C

                                       SCALAR MAP
 SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
 A1        000118        STNMX     00011C        I         000120        IJK       000124        K         000128
 INTL      00012C        E1        000130        E2        000134        A2        000138        E3        00013C
 A3        000140        J         000144        SIG       000148        NU12      00014C        NU21      000150
 XNU       000154        A4        000158        IDP       00015C

                                       ARRAY MAP
 SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
 STN       000160        STRAIN    00016C

                                       SUBPROGRAMS CALLED
 SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
 PLST      000170

                                       LABEL MAP
 LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION
   100     000326          105     000368          110     0003D8          120     00049C          130     000558
   135     000568          140     00057A          150     0005FA          160     000676          165     0006AE
   170     00068A          180     0006C6          200     000752          210     000798

  TOTAL MEMORY REQUIREMENTS 0007E0 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//PLST    EXEC FORTRAN(BCD,MAP)

  0001              FUNCTION PLST (SIG1,SIG2,STRN,DEL,J)
  0002              PLST = 1.0
  0003              IF (J-7) 100,120,120
  0004          100 IF (SIG2) 120,120,110
  0005          110 PLST = SIG1 + (SIG2-SIG1) * (ABS(STRN)-DEL*(J-1)) / DEL
  0006          120 CONTINUE
  0007              RETURN
  0008              END

                                       EQUIVALENCE DATA MAP
 SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
 PLST      0000E0

                                       SCALAR MAP
 SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION      SYMBOL    LOCATION
 J         0000E4        SIG2      0000E8        SIG1      0000EC        STRN      0000F0        DEL       0000F4

                                       LABEL MAP
 LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION      LABEL     LOCATION
   100     0001E6          110     0001F2          120     00023A

  TOTAL MEMORY REQUIREMENTS 00027C BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//BUKL    EXEC FORTRAN(BCD,MAP)
```

```
FORTRAN IV     MODEL 44  PS        VERSION 3, LEVEL 4   DATE  71321                    PAGE  0001

      SUBROUTINE BUKL (FBKL)
      COMMON / MATL / E(4,5),FXT(5),FYT(5),FXC(5),FYC(5),SSS(5),CMS(10)
      COMMON / ONE / SIDE,HT,THK(5),THETA,Y1OFF,Y2OFF,THT,DUM,TC(11)
     1  ,XC(11),HED(12),NX,NY,IOPT,ITYP,IFACE,IHOLE,ITET,ICNT,NNEW(4)
      DIMENSION FBKL(5),D(3),XL(5)
      N = 3
      IF (IFACE .EQ. 1) N = 5
      DO 500 I = 1,N
      XL(I) = HT
      XNU = 1. - E(4,I)*E(2,I)/E(1,I)*E(4,I)
      E12 = E(2,I)*E(4,I)/XNU
      TCUB  = THK(I)**3/12.
      D(1) = E(1,I)*TCUB/XNU
      D(2) = E(2,I)*TCUB/XNU
      D(3) = (E12    + E(3,I)*2.) *TCUB
      IF (IFACE .EQ. 1) GO TO 277
      IF (I-3) 280,280,277
  277 XNX = 2.*(3.14159/XL(I))**2*((D(1)*D(2))**.5 + D(3))
      GO TO 345
  280 PHA = 3.14159*HT/SIDE
      GAM = 4.*E(3,I)*XNU/E(2,I) + E(4,I)
      DPHA = D(3)/D(2)*PHA
      INO =  0
      ICOT = 0
      Z = 1.
      XNX = D(1)*(PHA/XL(I))**2 + 1.
      XFT = XNX/50.
  290 DPHB = (-D(1)*PHA**2 + XNX*XL(I)**2) / D(2)
      BETA = PHA**.5*((DPHA**2+DPHB)**.5-DPHA)**.5
      ALPA = PHA**.5*((DPHA**2+DPHB)**.5+DPHA)**.5
      XA = BETA*(BETA**2+PHA**2*GAM)*(ALPA**2-PHA**2*.3)*TANH(ALPA)
      XB = ALPA*(ALPA**2-PHA**2*GAM)*(BETA**2+PHA**2*.3)*TAN(BETA)
      ZL = Z
      Z = XA - XB
      SGN = Z/ZL
      IF (SGN) 292,292,300
  292 IF (INO) 293,300,293
  293 IF (ICOT - 3) 295,295,345
  295 XFT = XFT/10.
      XNX = XNXL
      Z = ZL
      ICOT = ICOT + 1
  300 XNXL = XNX
      XNX = XNX + XFT
      INO = INO + 1
      GO TO 290
  345 FBKL(I) = XNX/THK(I)
  500 CONTINUE
      IF (ICNT .GT. 1 .AND. IOPT .EQ. 1) RETURN

      WRITE (6,6000) (FBKL(I),I=1,N)
 6000 FORMAT ( 1H1, 21H  LEG BUCKLING STRESS ,/, 5F10.0 )
      RETURN
      END
```

```
                                   COMMON BLOCK / MATL    / MAP SIZE  0000DC
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
E         000000       FXT       000050       FYT       000064       FXC       000078       FYC       00008C
SSS       0000A0       CNS       000084

                                   COMMON BLOCK / ONE     / MAP SIZE  0000E8
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
SIDE      000000       HT        000004       THK       000008       THETA     00001C       Y1OFF     000020
Y2OFF     000024       THT       000028       DUM       00002C       TC        000030       XC        00005C
HED       000038       NX        000088       NY        00008C       IOPT      0000C0       ITYP      0000C4
IFACE     0000C8       IHOLE     0000CC       ITET      0000D0       ICNT      0000D4       NNEW      0000D8

                                   SCALAR MAP
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
N         000110       I         000114       XNU       000118       E12       00011C       TCUB      000120
XNX       000124       PHA       000128       GAM       00012C       DPHA      000130       INO       000134
ICCT      000138       Z         00013C       XFT       000140       DPHB      000144       BETA      000148
ALPA      00014C       XA        000150       XB        000154       ZL        000158       SGN       00015C
XNXL      000160

                                   ARRAY MAP
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
FBKL      000154       D         000168       XL        000174

                                   SUBPROGRAMS CALLED
SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION     SYMBOL    LOCATION
IBCOM#    000188       FRXPR#    00018C       TANH      000190       TAN       000194

                                   LABEL MAP
LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION     LABEL     LOCATION
  277     000322         280     000368         290     0003FE         292     000558         293     000564
  295     000578         300     0005A4         345     0005CE         500     0005F2        6000     000694

 TOTAL MEMORY REQUIREMENTS 000734 BYTES

COMPILER HIGHEST SEVERITY CODE WAS  0
//SYSABL ACCESS SDSABS,090='SYSRET'
// DELETE SDSABS(TETRA,TETRA1,TETRA2,TETRA3,TETRA4,TETRA5,TETRA6)
// DELETE SDSABS(TETRA7,TETRA8,TETRA9,TETRA10,TETRA11,TETRA12)
// CONDENSE SDSABS
//       EXEC LNKEDT(KEEP)

 71/321                                    JC710280 LINKAGE EDIT

LIST    PHASE TETRA,ROOT
LIST    INCLUDE TETMN,L
LIST    INCLUDE OUTIN,L
LIST    INCLUDE CURV,L
LIST   AUTOLINK     DIOCS#
LIST   AUTOLINK     LOAD
LIST   AUTOLINK     IBCOM#
LIST   AUTOLINK     ATAN2
LIST   AUTOLINK     VDIOCS#
LIST   AUTOLINK     UNITAB#
LIST   AUTOLINK     USEROPT
LIST    PHASE TETRA1,*
LIST    INCLUDE GEN,L
```

```
LIST     INCLUDE OPTIM,L
LIST     INCLUDE WEIGHT,L
LIST     INCLUDE DAR,L
LIST     INCLUDE NONLIN,L
LIST    AUTOLINK     EXIT
LIST    AUTOLINK     FRXPI0
LIST    AUTOLINK     SQRT
LIST    AUTOLINK     FRXPR0
LIST    AUTOLINK     ALOG
LIST    AUTOLINK     EXP
LIST     PHASE TETRA2,*
LIST     INCLUDE NODGEN,L
LIST    AUTOLINK     SIN
LIST     PHASE TETRA3,TETRA2
LIST     INCLUDE PLATGN,L
LIST     PHASE TETRA4,TETRA2
LIST     INCLUDE LODGEN,L
LIST    AUTOLINK     SQRT
LIST    AUTOLINK     SIN
LIST     PHASE TETRA5,TETRA2
LIST     INCLUDE NODGAN,L
LIST    AUTOLINK     SIN
LIST    AUTOLINK     TAN
LIST     PHASE TETRA6,TETRA2
LIST     INCLUDE PLATGA,L
LIST     PHASE TETRA7,TETRA2
LIST     INCLUDE LODGAN,L
LIST    AUTOLINK     SQRT
LIST    AUTOLINK     SIN
LIST     PHASE TETRA8,TETRA1
LIST     INCLUDE STRSS,L
LIST     PHASE TETRA9,*
LIST     INCLUDE STRESS,L
LIST     INCLUDE NID,L
LIST    AUTOLINK     MAXO
LIST    AUTOLINK     AMIN1
LIST     PHASE TETRA10,TETRA9
LIST     INCLUDE STIFF,L
LIST     INCLUDE TRING,L
LIST     INCLUDE TRPRD,*
  71/321                                   JC710200 LINKAGE EDIT

LIST    AUTOLINK     SQRT
LIST     PHASE TETRA11,TETRA1
LIST     INCLUDE SOLPAC,L
LIST    AUTOLINK     SQRT
LIST     PHASE TETRA12,TETRA1
LIST     INCLUDE DEFL,L
LIST     INCLUDE MARGIN,L
LIST     INCLUDE MODUL,L
LIST     INCLUDE BUKL,L
LIST     INCLUDE PLST,L
LIST    AUTOLINK     FRXPR0
LIST    AUTOLINK     SQRT
LIST    AUTOLINK     TANH
LIST    AUTOLINK     TAN
LIST    AUTOLINK     ALOG
LIST    AUTOLINK     EXP
LIST        ENTRY                                            CREATED
```

71/321	PHASE	TRANSFER ADDR.	LOCORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
COMMON						COMMON	COMN	0041D0	00006C
COMMON						COMMON	MATL	004240	0000DC
COMMON						COMMON	LODA	004320	000024
COMMON						COMMON	ONE	004348	0000E8
COMMON						COMMON	OPTA	004430	000284
COMMON						COMMON	MODU	0046E8	000550
COMMON						COMMON	LODB	004C38	000168
COMMON						COMMON	RAX	004DA0	000024
COMMON						COMMON	DUMP	004DC8	000004
COMMON						COMMON	PLAT	004DD0	0044E0
COMMON						COMMON	LODS	009280	0012C0
COMMON						COMMON	XYZ	00A570	003CF4
COMMON						COMMON	ELM	00E268	0008A0
ROOT	TETRA	00E808		00E808	012A5F	686	CSECT	MAIN446	00E808	00E808
						* ENTRY	MAIN44	00E808	
						CSECT	OUTINK	00EE60	00EE60
						ENTRY	OUTIN	00EE60	
						CSECT	CURVE	00F0D0	00F0D0
						ENTRY	CURV	00F0D0	
						CSECT	BOADIOCS	00F380	00F380
						ENTRY	DIOCS#	00F380	
						CSECT	BOADVLY	00F670	00F670
						ENTRY	LOAD	00F688	
						* ENTRY	LINK	00F678	
						CSECT	BOAIBCOM	00F788	00F788
						ENTRY	IBCOM#	00F788	
						ENTRY	FIRSTIM	010BE8	
						ENTRY	FIXPOINT	010BB0	
						* ENTRY	ADCON#	00F844	
						CSECT	BOASATN2	012110	012110
						ENTRY	ATAN2	012116	
						* ENTRY	ATAN	012130	
						CSECT	BOAFIOCS	012288	012288
						ENTRY	VDIOCS#	01299C	
						ENTRY	BUFORG#	012994	
						ENTRY	RCBORG#	012998	
						ENTRY	FIOCD#	0122C2	
						ENTRY	FIOCS#	012288	

						CSECT	BOAUNITB	012A08	012A08
						ENTRY	UNITAB#	012A08	
						CSECT	BOAUOPT	012A58	012A58
						ENTRY	USEROPT	012A58	
	TETRA1	012A60	012A60	01787F	709	CSECT	GEN&	012A60	012A60
						ENTRY	GEN	012A60	
						CSECT	OPTM&	015280	015280
						ENTRY	OPTM	015280	
						CSECT	WEIGHT&	016898	016898
						ENTRY	WEIGHT	016898	
						CSECT	DAR&	016FA8	016FA8
						ENTRY	DAR	016FA8	
						CSECT	NONLIN&	017320	017320
						ENTRY	NONLIN	017320	
						CSECT	BOAFEXIT	0176A0	0176A0
						ENTRY	EXIT	0176A6	
						CSECT	BOAFRXPI	0176C0	0176CC
						ENTRY	FRXPI#	0176C8	
						CSECT	BOASSQRT	017778	017778
						ENTRY	SQRT	01777E	
						CSECT	BOAFRXPR	017828	017828
						ENTRY	FRXPR#	017830	
						CSECT	BOASLOG	017920	017920
						ENTRY	ALOG	017942	
						* ENTRY	ALOG10	017928	
						CSECT	BOASEXP	017A30	017A30
						ENTRY	EXP	017A34	
	TETRA2	017B80	017B80	018BB7	738	CSECT	NODGEN&	017B80	017B80
						ENTRY	NODGEN	017B80	
						CSECT	BOASSCN	0187A0	0187A0
						ENTRY	SIN	0187C0	
						ENTRY	COS	0187A4	
	TETRA3	017B80	017B80	019607	743	CSECT	PLATGN&	017B80	017B80
71/321	PHASE	TRANSFER ADDR.	LOCORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
						ENTRY	PLATGN	017B80	
	TETRA4	017B80	017B80	019BD7	753	CSECT	LUDGEN&	017B80	017B80
						ENTRY	LODGEN	017B80	
						CSECT	BOASSQRT	019A10	019A10
						ENTRY	SQRT	019A16	

						CSECT	BOASSCN	019AC0	019AC0
						ENTRY	SIN	019AE0	
						ENTRY	COS	019AC4	
	TETRA5	017B80	017B80	018A37	765	CSECT	NODGANG	017B80	017B80
						ENTRY	NODGAN	017B80	
						CSECT	BOASSCN	0187F0	0187F0
						ENTRY	SIN	018810	
						ENTRY	COS	0187F4	
						CSECT	BOASTNCT	018908	018908
						ENTRY	TAN	018926	
						* ENTRY	COTAN	01890E	
						* ENTRY	QTAN	018A34	
	TETRA6	017B80	017B80	01911F	771	CSECT	PLATGAG	017B80	017B80
						ENTRY	PLATGA	017B80	
	TETRA7	017B80	017B80	01A84F	779	CSECT	LODGANG	017B80	017B80
						ENTRY	LODGAN	017B80	
						CSECT	BOASSQRT	01A688	01A688
						ENTRY	SQRT	01A68E	
						CSECT	BOASSCN	01A738	01A738
						ENTRY	SIN	01A758	
						ENTRY	COS	01A73C	
	TETRA8	012A60	012A60	012BEF	795	CSECT	STRSSG	012A60	012A60
						ENTRY	STRSS	012A60	
	TETRA9	012BF0	012BF0	017E6F	796	CSECT	STRESSG	012BF0	012BF0
						ENTRY	STRESS	012BF0	
						CSECT	NIDG	017A50	017A50
						ENTRY	NID	017A50	
						CSECT	BOAFMAXI	017C30	017C30
						ENTRY	MAXO	017C36	
						ENTRY	MINO	017C5C	
						* ENTRY	AMINO	017CA8	
						* ENTRY	AMAXO	017C82	
						CSECT	BOAFMAXR	017D50	017D50
71/321	PHASE	TRANSFER ADDR.	LOCORE	HICORE	BLOCK NO.	ESD TYPE	LABEL	LOADED	REL-FACTOR
						ENTRY	AMIN1	017DC8	
						* ENTRY	AMAX1	017DA2	
						* ENTRY	MIN1	017D7C	
						* ENTRY	MAX1	017D56	
	TETRA10	012BF0	012BF0	01C627	826	CSECT	STIFFG	012BF0	012BF0
						ENTRY	STIFF	012BF0	
						CSECT	TRIM6G	017F58	017F58
						ENTRY	TRIM6	017F58	

```
                                                                   CSECT      TRPRDG      018BD0      018BD0
                                                                     ENTRY    TRPRD       018BD0

                                                                   CSECT      BOASSORT    01C578      01C578
                                                                     ENTRY    SORT        01C57E

         TETRA11    012A60      012A60    01AA6F      881          CSECT      SOLPACG     012A60      012A60
                                                                     ENTRY    SOLPAC      012A60

                                                                   CSECT      BOASSORT    01A9C0      01A9C0
                                                                     ENTRY    SORT        01A9C6

         TETRA12    012A60      012A60    01709F      927          CSECT      DEFLG       012A60      012A60
                                                                     ENTRY    DEFL        012A60

                                                                   CSECT      MARGING     0153D8      0153D8
                                                                     ENTRY    MARGIN      0153D8

                                                                   CSECT      MODULG      0158C0      0158C0
                                                                     ENTRY    MODUL       0158C0

                                                                   CSECT      BUKLG       0160A0      0160A0
                                                                     ENTRY    BUKL        0160A0

                                                                   CSECT      PLSTG       0167D8      0167D8
                                                                     ENTRY    PLST        0167D8

                                                                   CSECT      BOAFRXPR    016A58      016A58
                                                                     ENTRY    FRXPR8      016A60

                                                                   CSECT      BOASSORT    016B50      016B50
                                                                     ENTRY    SORT        016B56

                                                                   CSECT      BOASTANH    016C00      016C00
                                                                     ENTRY    TANH        016C06

                                                                   CSECT      BOASTNCT    016D10      016D10
                                                                     ENTRY    TAN         016D2E
                                                                   * ENTRY    COTAN       016D16
                                                                   * ENTRY    QTAN        016E3C

                                                                   CSECT      BOASLOG     016E40      016E40

  71/321   PHASE    TRANSFER ADDR.  LOCORE    HICORE    BLOCK NO.  ESD TYPE   LABEL       LOADED      REL-FACTOR

                                                                     ENTRY    ALOG        016E62
                                                                   * ENTRY    ALOG10      016E48

                                                                   CSECT      BOASEXP     016F50      016F50
                                                                     ENTRY    EXP         016F54

LINKAGE EDITOR HIGHEST SEVERITY WAS  0
/&
   142803
```

LITERATURE CITED

1. Tocher, J. L., AREA COORDINATES AND THEIR USE IN FINITE ELEMENT DERIVATIONS, Boeing Co. Internal Report MA-93. May 1967, pp 3-5, 17-19, 26-29.

2. Felippa, C. A., REFINED FINITE ELEMENT ANALYSIS OF LINEAR AND NONLINEAR STRUCTURES, Structures and Materials Report SESM 66-22. Dept. Civil Engineering, University of California at Berkeley, October 1966.

3. Przemieniecki, J. S., THEORY OF MATRIX STRUCTURAL ANALYSIS, McGraw-Hill, 1968, pp 52-64, 83-89, 433-437

4. Gellatly, R. A., Gallagher, R. H., and Luberacki, W. A., DEVELOPMENT OF A PROCEDURE FOR AUTOMATED SYNTHESIS OF MINIMUM-WEIGHT STRUCTURES, AFFDL-TDR-64-141.

5. Petit, P. H., and Waddoups, M. E., A METHOD OF PREDICTING THE NONLINEAR BEHAVIOR OF LAMINATED COMPOSITES, J. Composite Materials, Vol. 3, Jan 1969, pp 2-19.

6. Lackmann, L. M., and Ault, R. M., MINIMUM-WEIGHT ANALYSIS OF FILAMENTARY COMPOSITE WIDE COLUMNS, J. Aircraft, Vol. 5, No. 2, March-April 1960, pp 184-190.

7. Timoshemko, S., and Gere, J. M., THEORY OF ELASTIC STABILITY, New York, McGraw-Hill 1961, pp 403-404.

8. Tsai, S. W., Adams, D. F., and Doner, D. R., ANALYSIS OF COMPOSITE STRUCTURES, NASA Report CR-620, November 1966, pp 3-5.

GLOSSARY

A	- Merged stiffness matrix
$\bar{A}$	- Triangularized stiffness matrix
B	- Matrix relating plate strains to plate nodal deflections
C_{11}, C_{22}, etc	- Plate elasticity matrix
D_1, D_2, D_3	- Plate bending stiffnesses
E_x	- Plate elastic modulus in fiber direction
E_y	- Plate elastic modulus transverse to fiber direction
F_{XT}	- Plate tension allowable stress in fiber direction
F_{XC}	- Plate compression allowable stress in fiber direction
F_{YT}	- Plate tension allowable stress transverse to fiber direction
F_{YC}	- Plate compression allowable stress transverse to fiber direction
G_{XY}	- Plate shear modulus
L_X	- Number of Vertical legs on X face
L_Y	- Number of Skew B legs on Y face
N_X	- Plate buckling load in X direction (lb/in.)
P_X	- Vertex nodal load for uniform load in X direction
P_Y	- Vertex nodal load for uniform load in Y direction
P_{XY}	- Vertex nodal load for uniform load in XY direction
P_{MX}	- Vertex nodal load for uniform moment in X direction

P_{MY} - Vertex nodal load for uniform moment in Y direction

P_{MXY} - Vertex nodal load for uniform moment in XY direction

PZ - Vertex nodal load in Z direction

APPENDIX I
ADDITIONAL GENERATION EXAMPLES

Additional examples of flat plate model generation are included in this appendix to clarify the examples given in the body of the report. The nodal numbering systems used for a 6 by 6 and a 10 x 6 true Tetra-core plate are shown in Figures 29 and 30, respectively. The nodal numbering systems used for a 4 x 6 and an 8 x 4 truncated Tetra-core plate are shown in Figures 31 and 32. Plate numbering systems used for the Vertical, Skew A, and Skew B legs in the truncated Tetra-core flat plate are shown in Figures 33, 34, and 35. The addition plates added to Skew A and Skew B legs to complete the cylinder model are shown in Figures 36 and 37. The nodal loads generated for a true Tetra-core flat plate model for each load type are shown in Figures 38, 39, 40, 41, 42, 43, 44, and 45. The nodal loads generated for a truncated Tetra-core flat plate model are shown in Figures 46, 47, 48, 49, and 50.

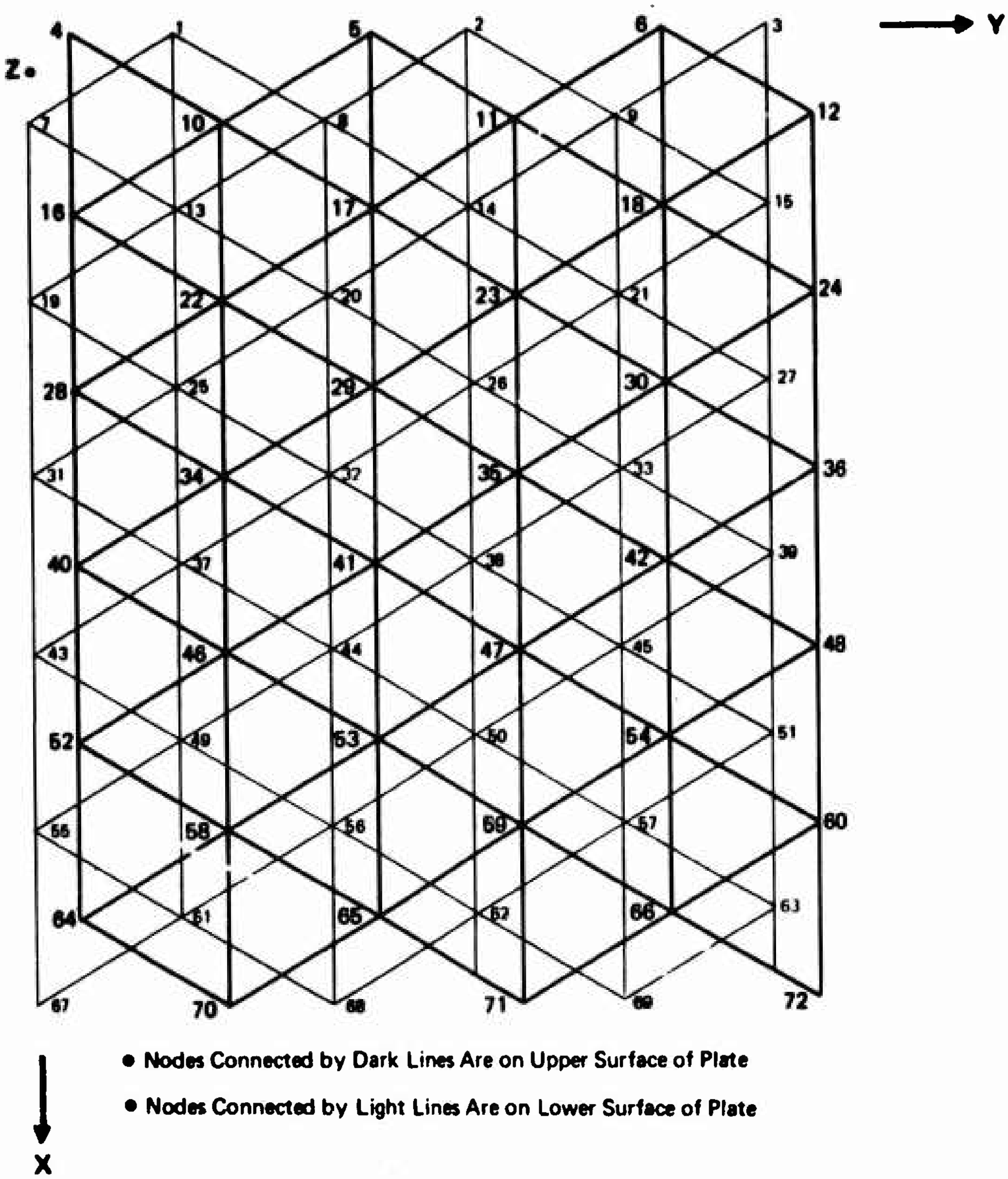


Figure 29. Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 6, L_Y = 6$.

Nodes Connected by Dark Lines Are on Upper Surface of Plate

Nodes Connected by Light Lines Are on Lower Surface of Plate

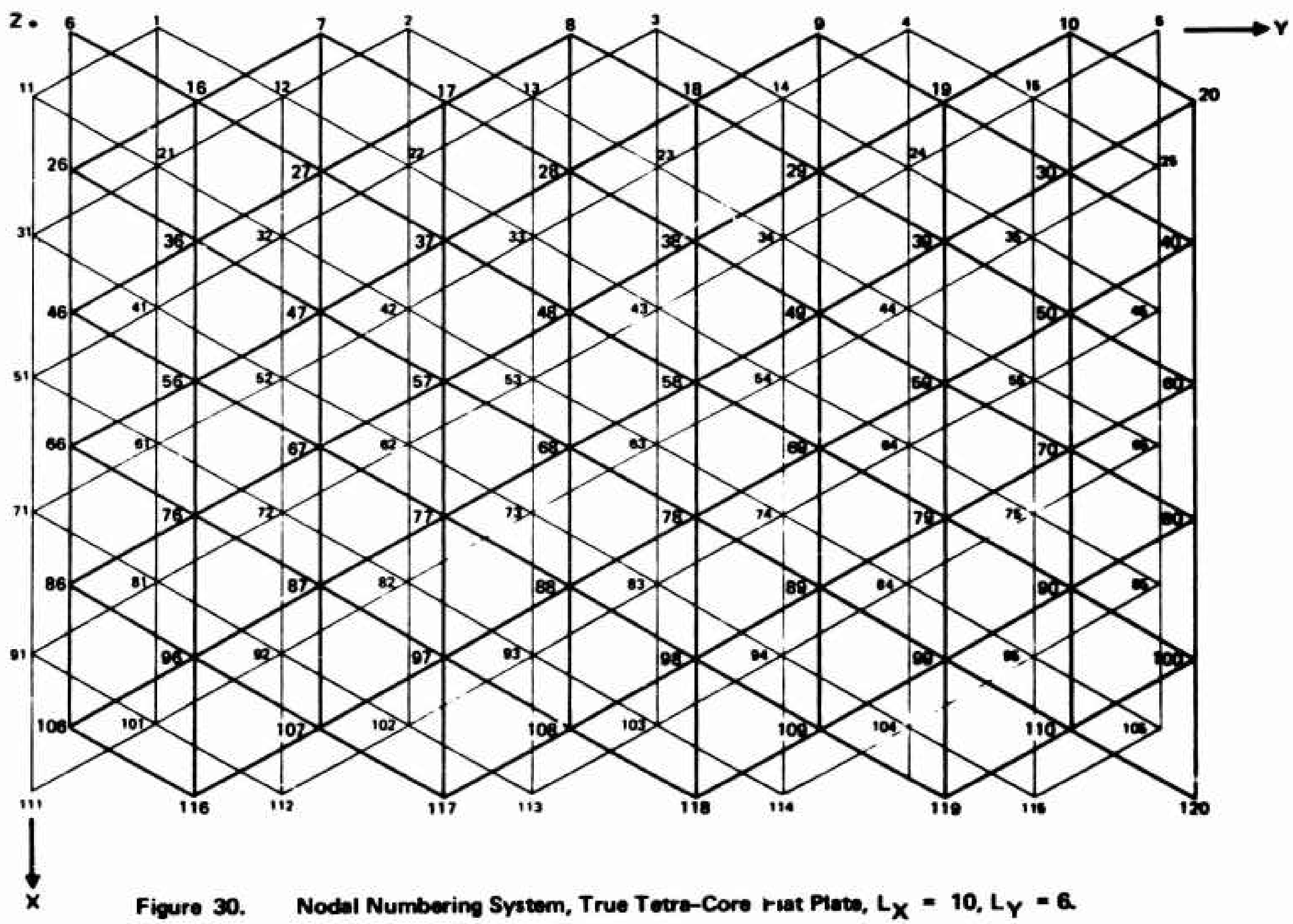


Figure 30. Nodal Numbering System, True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$.

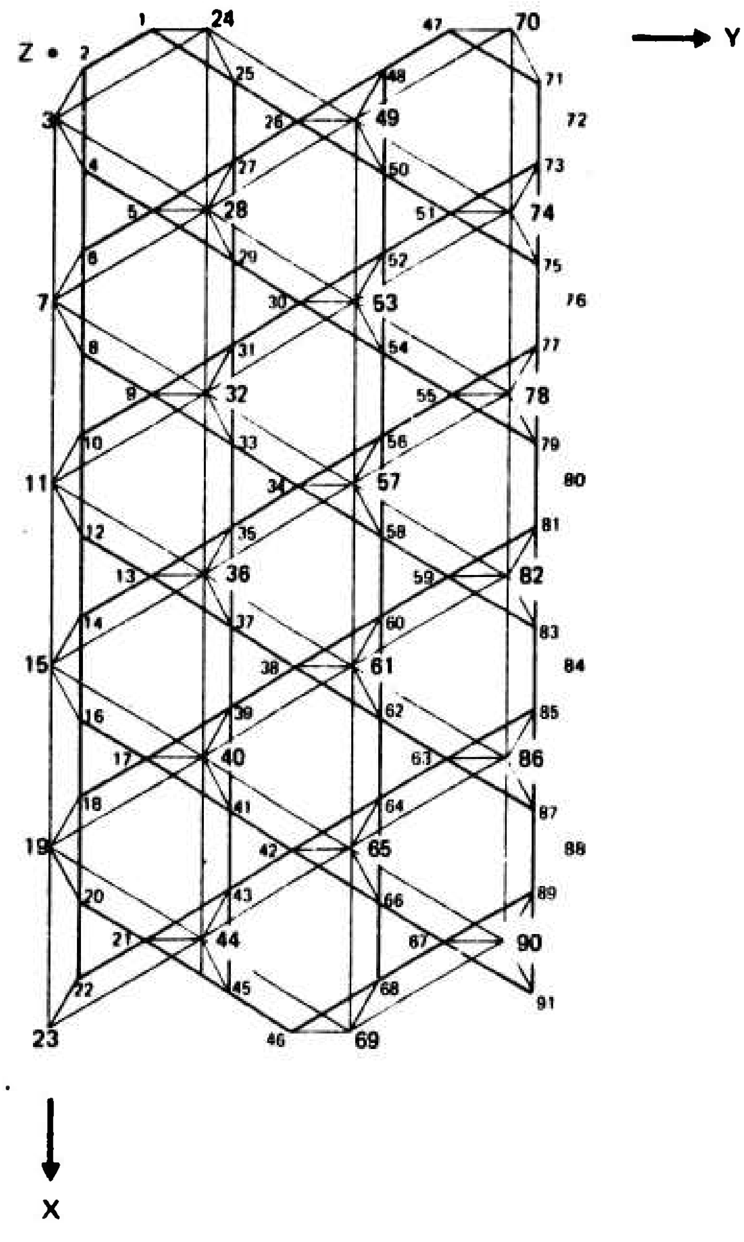


Figure 31. Nodal Numbering System, Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 6$.

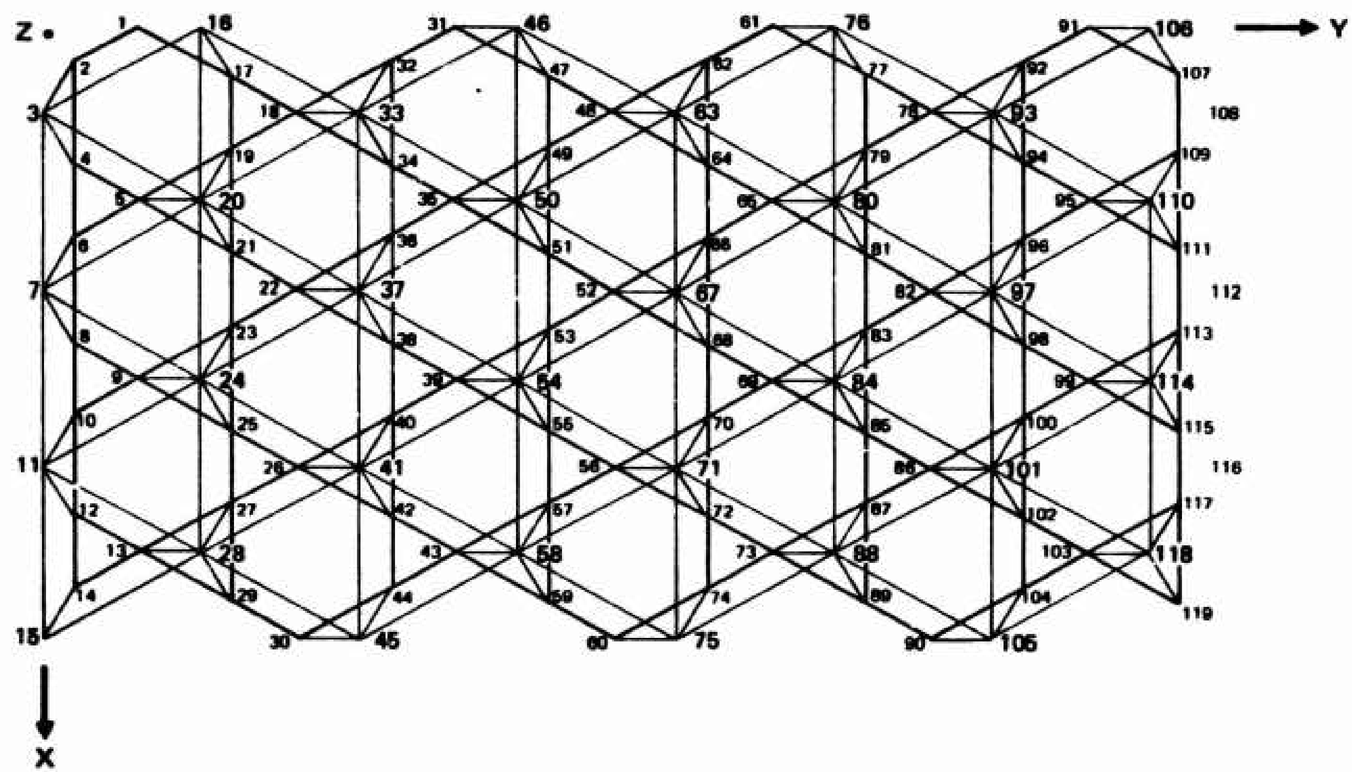


Figure 32. Nodal Numbering System, Truncated Tetra-Core Flat Plate, $L_X = 8, L_Y = 4$.

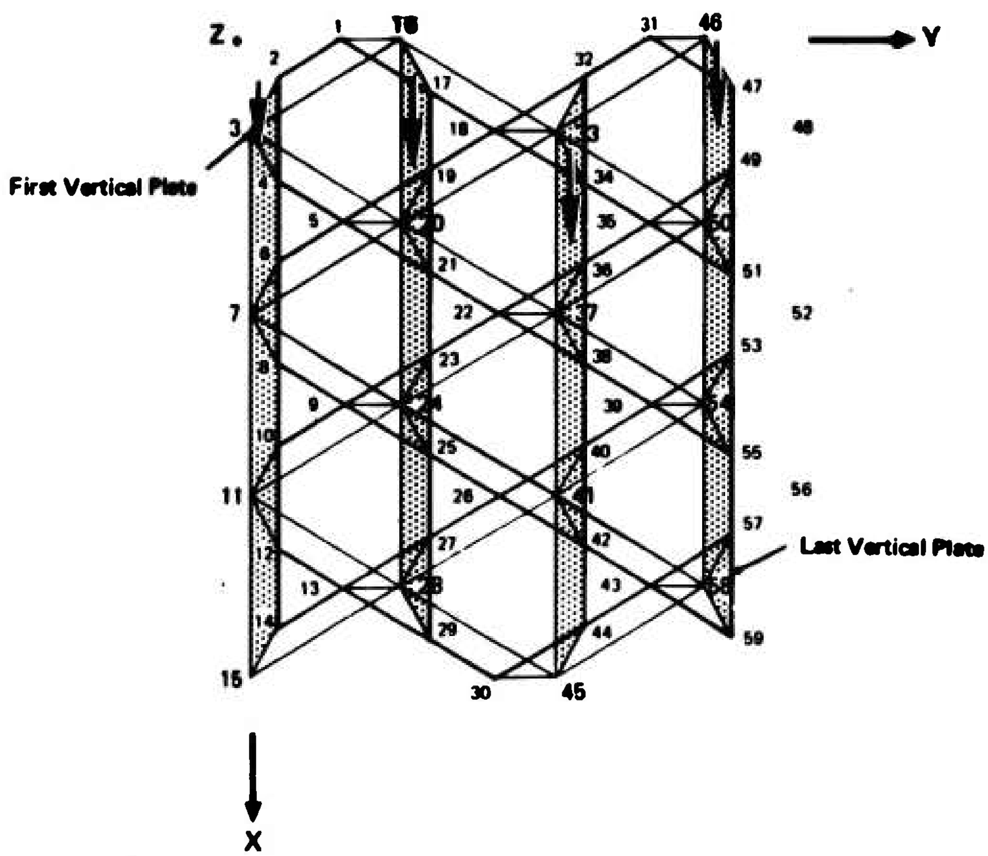


Figure 33. Vertical Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4, L_Y = 4$.

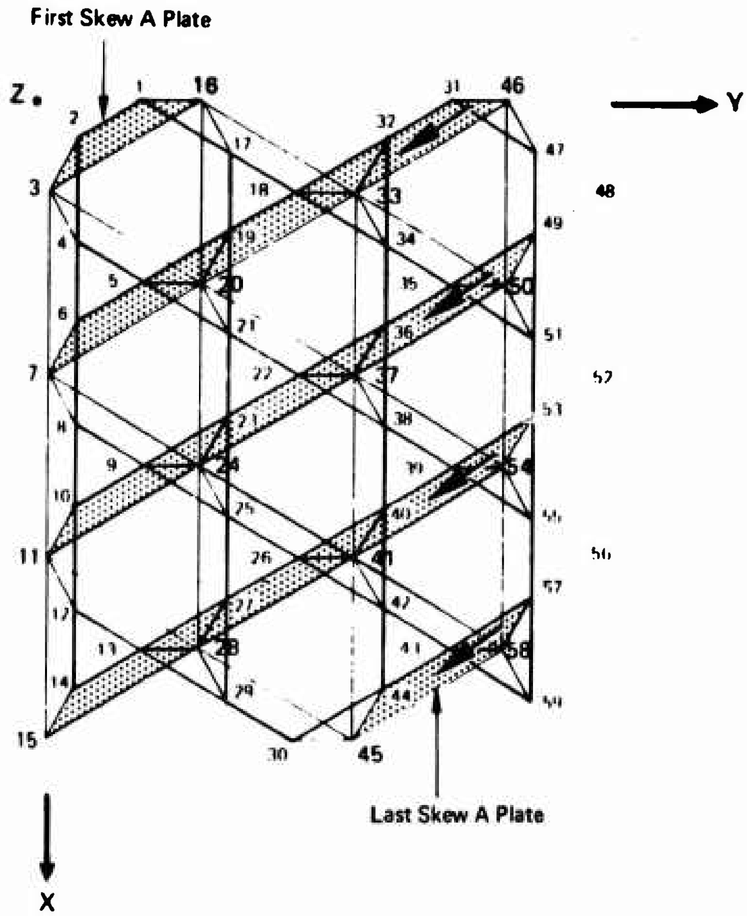


Figure 34. **Skew A Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4$, $L_Y = 4$.**

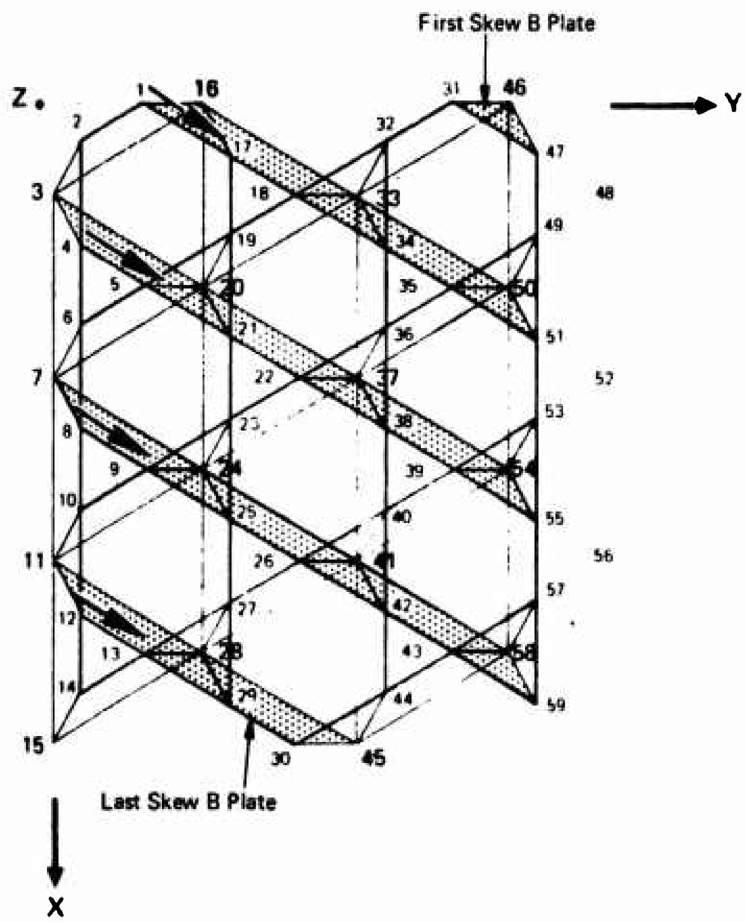


Figure 36. Skew B Leg Plate Numbering System Truncated Tetra-Core, $L_X = 4$, $L_Y = 4$.

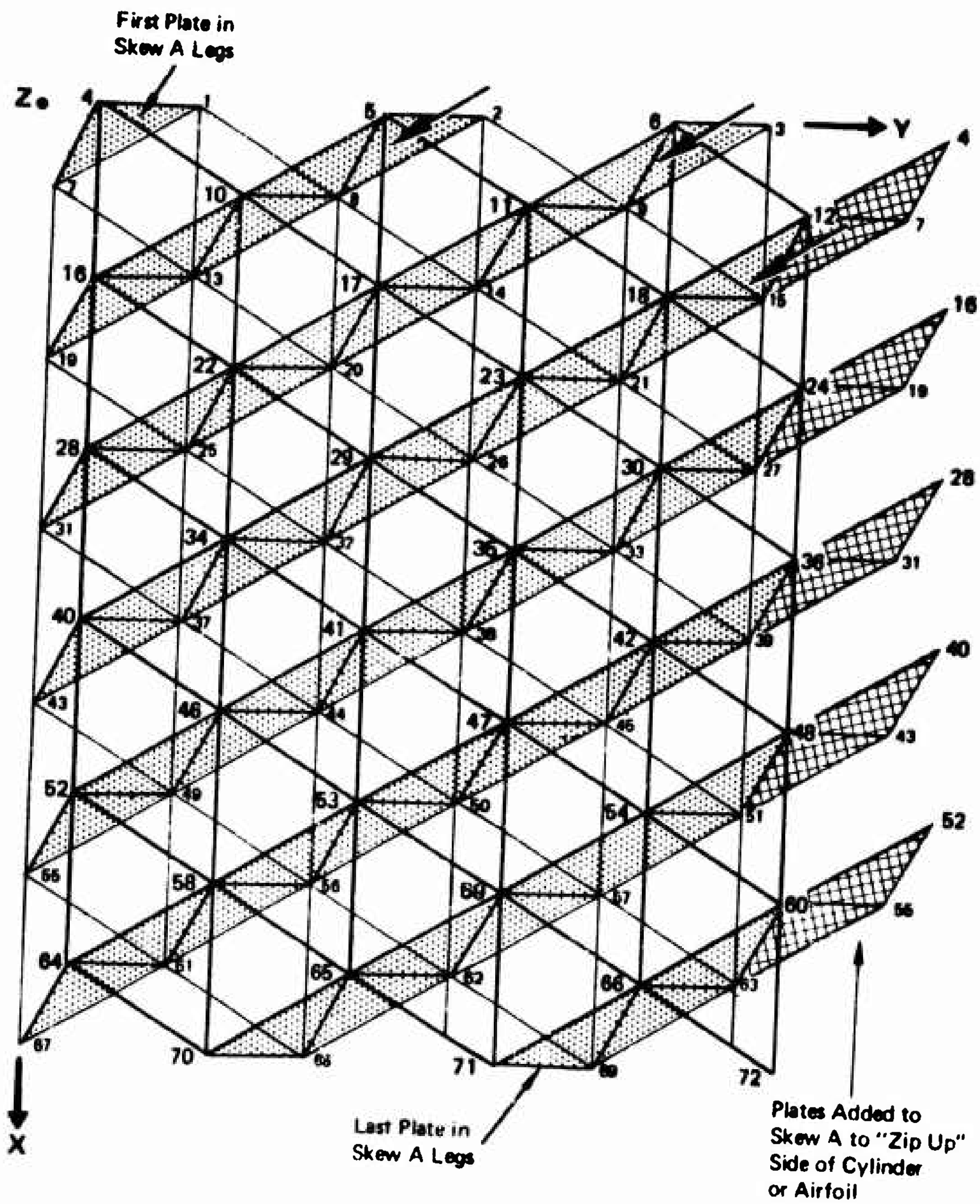


Figure 36. Skew A Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$.

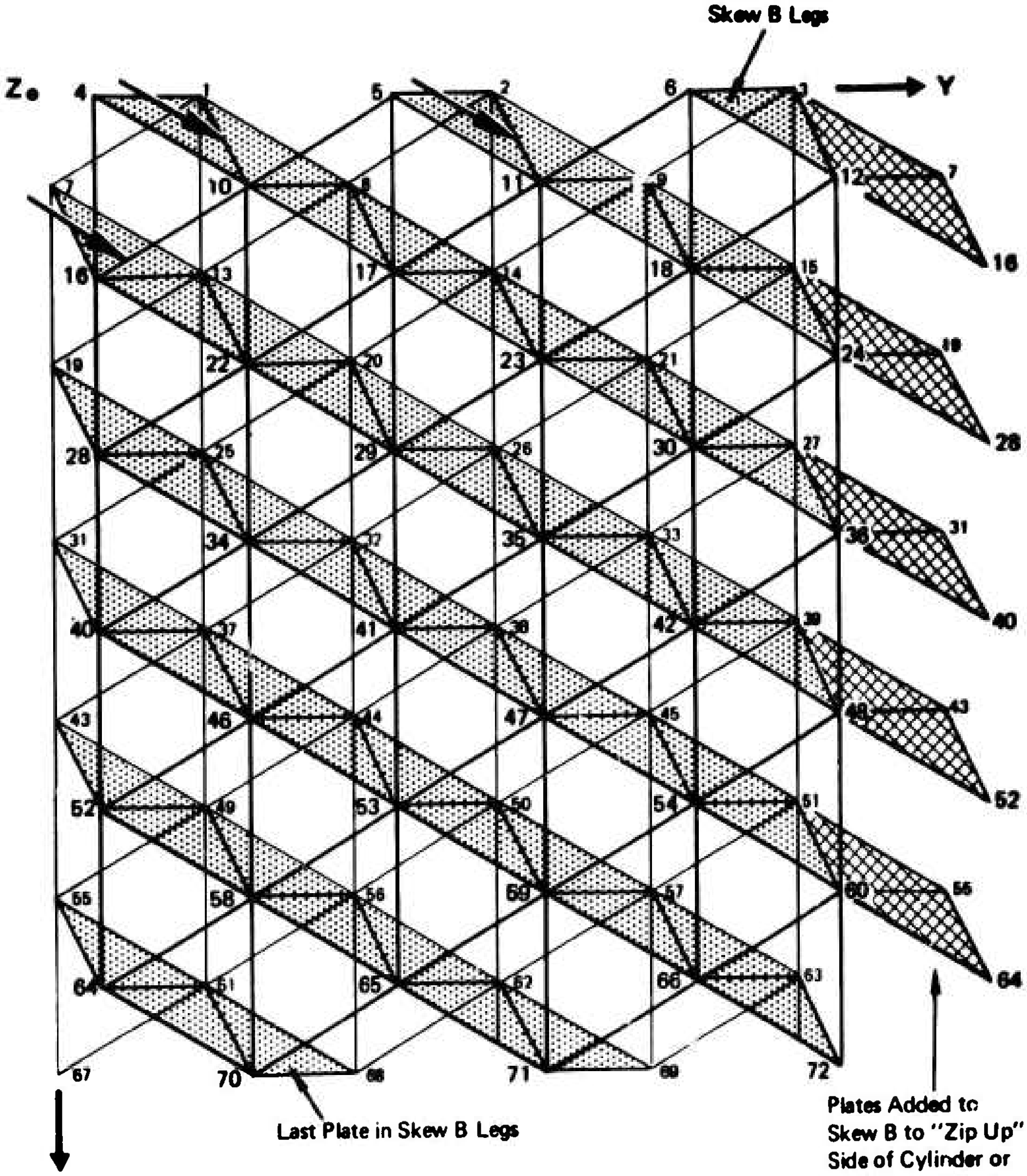


Figure 37. Skew B Plate Numbering System, True Tetra-Core Flat Plate, $L_X = 6$, $L_Y = 6$.

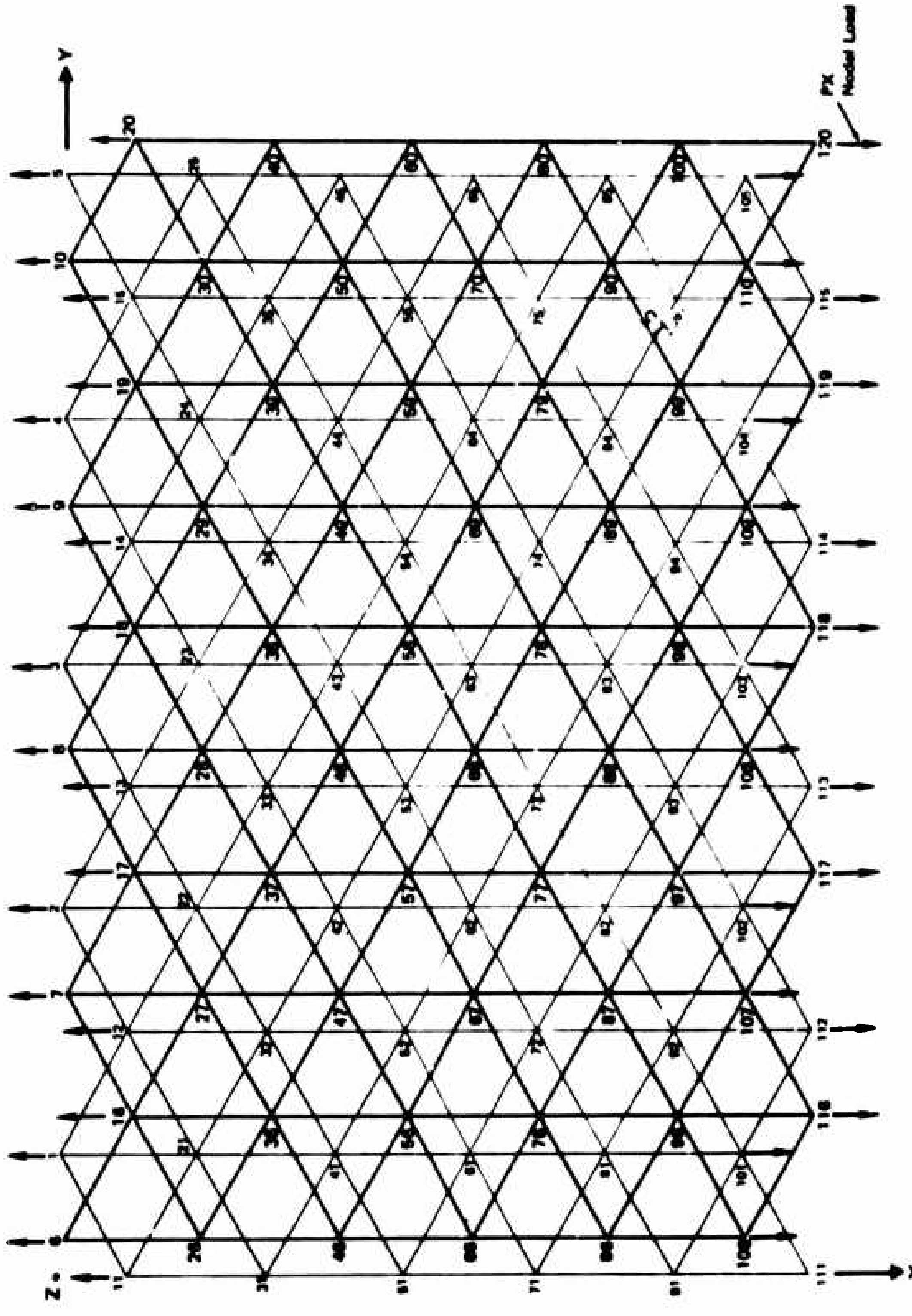


Figure 38. Load N_X Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$.

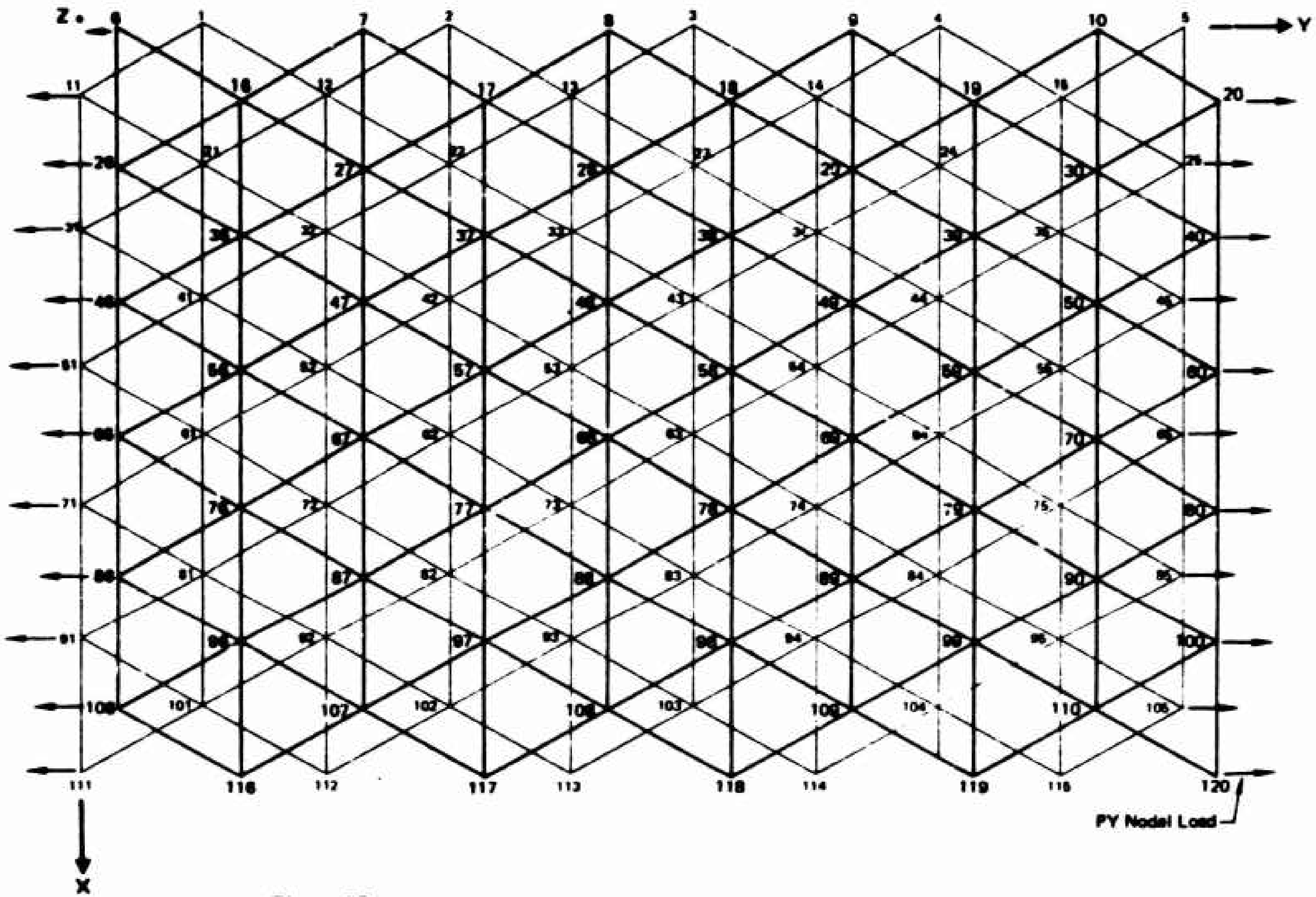


Figure 39. Load N_Y Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$.

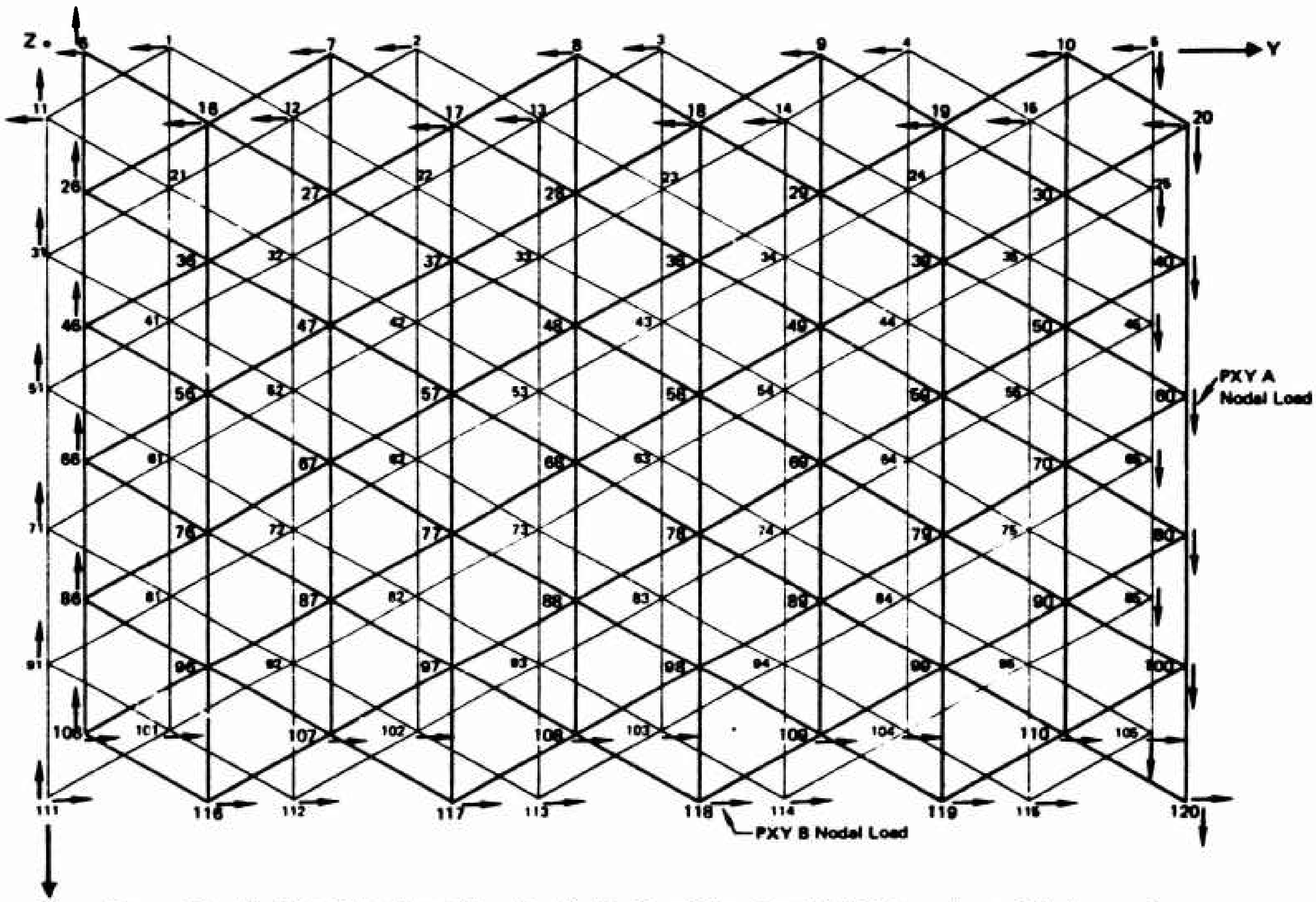


Figure 40. In–Plane Shear Load N_{XY} Applied to True Tetra–Core Flat Plate, $L_X = 10$, $L_Y = 6$.

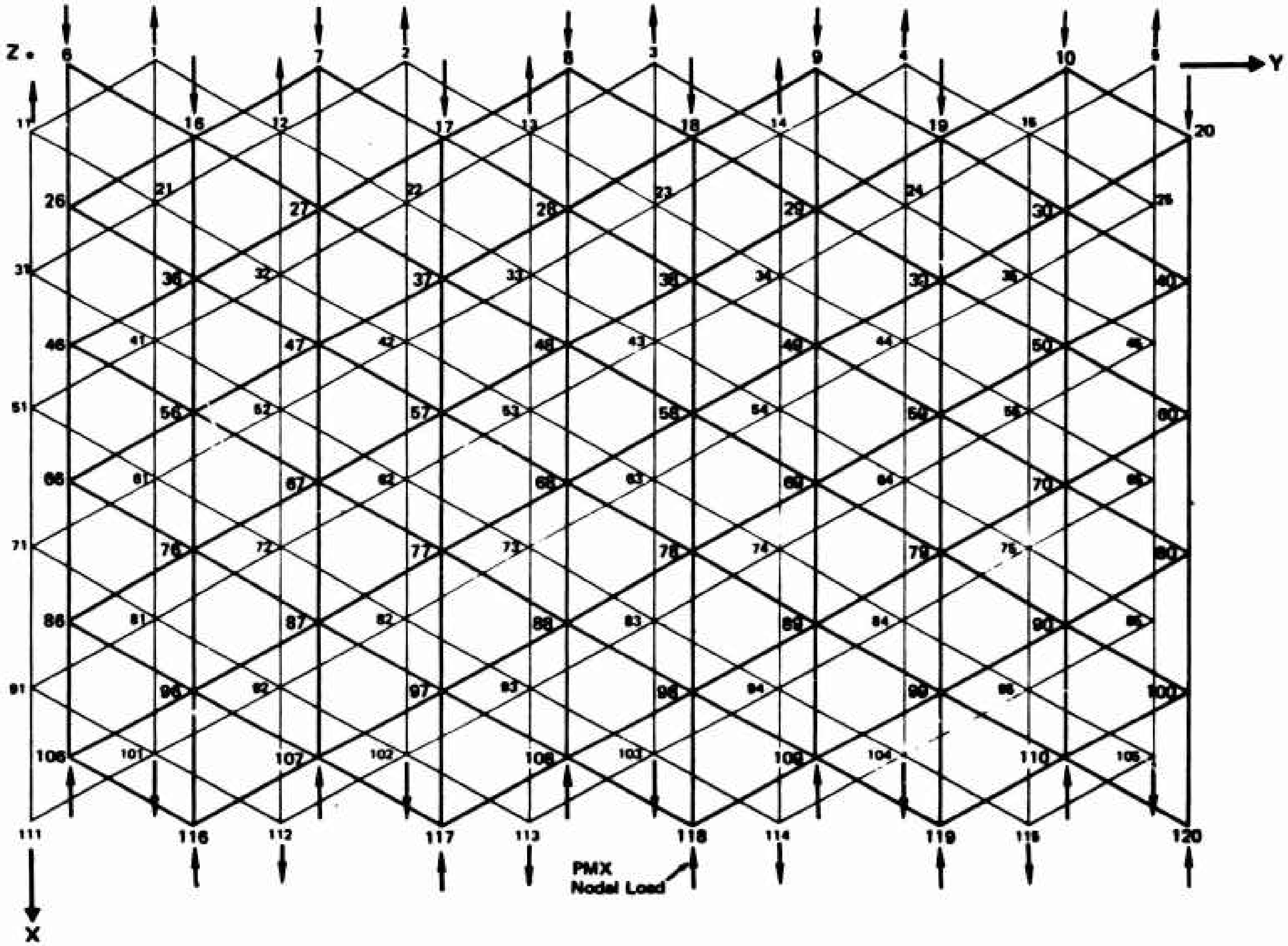


Figure 41. Moment M_X Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$.

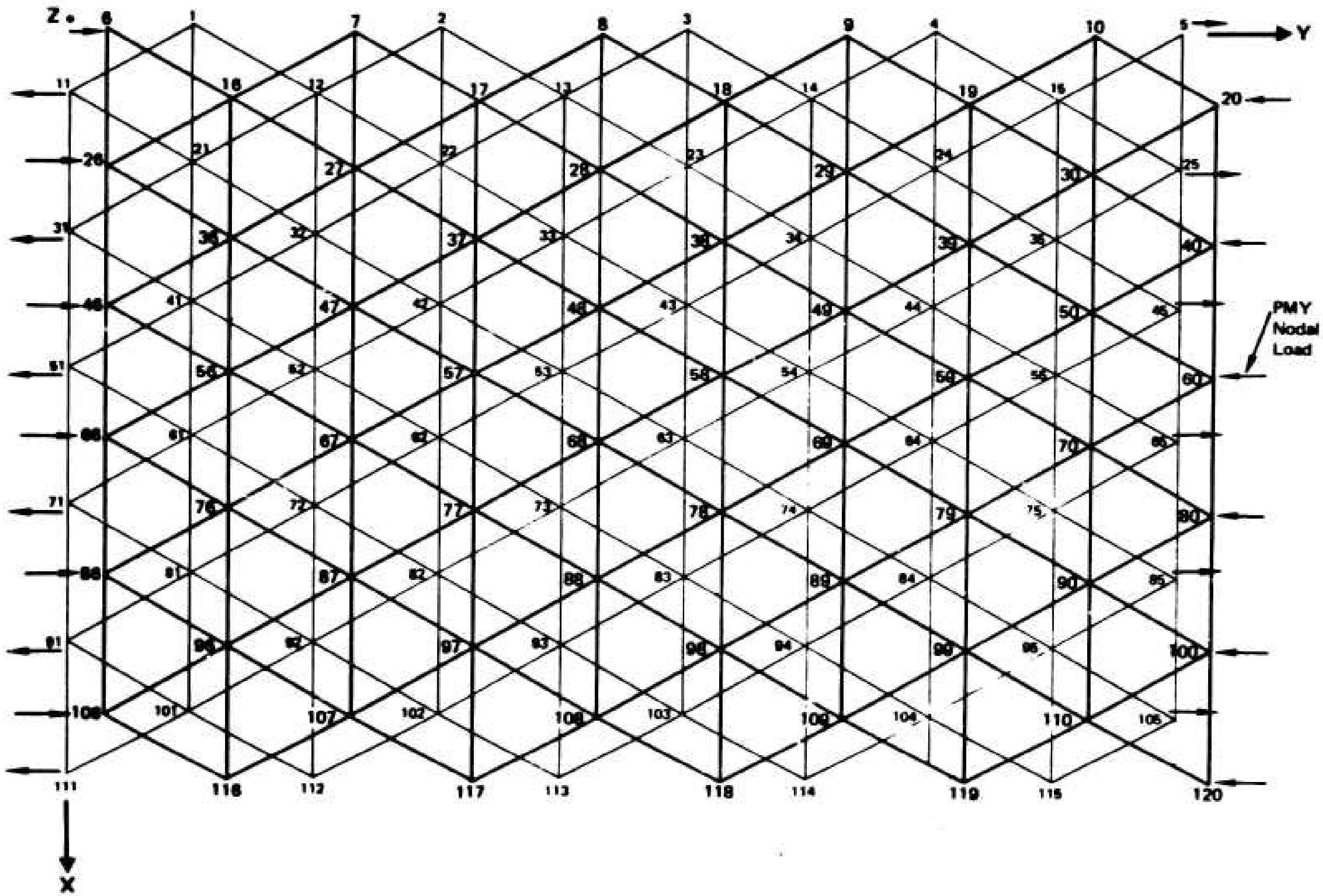


Figure 6.2 Moment M_Y Applied to True Tetra–Core Flat Plate, $L_X = 10$, $L_Y = 6$.

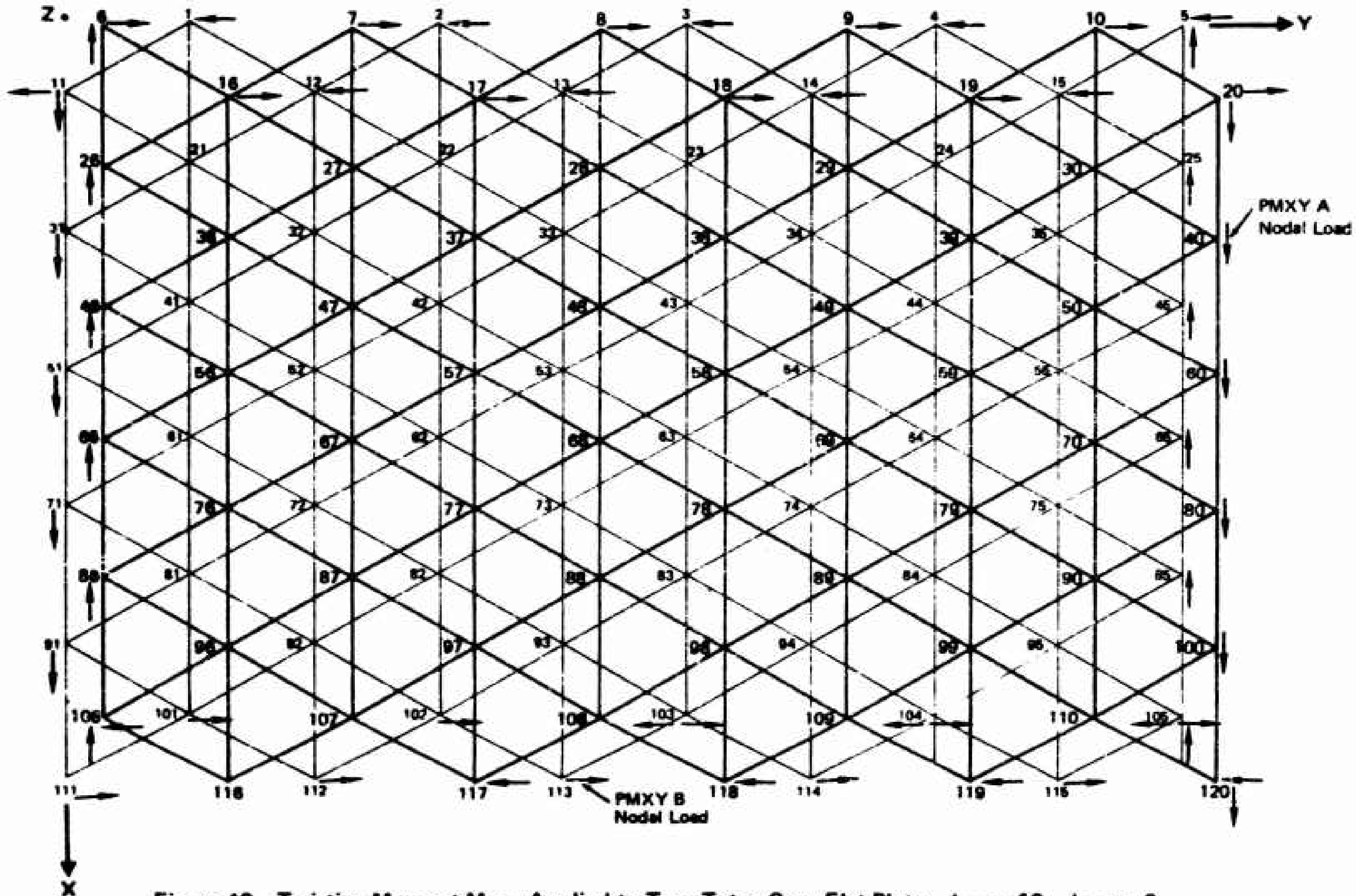


Figure 43. Twisting Moment M_{XY} Applied to True Tetra-Core Flat Plate, $L_X = 10$, $L_Y = 6$.

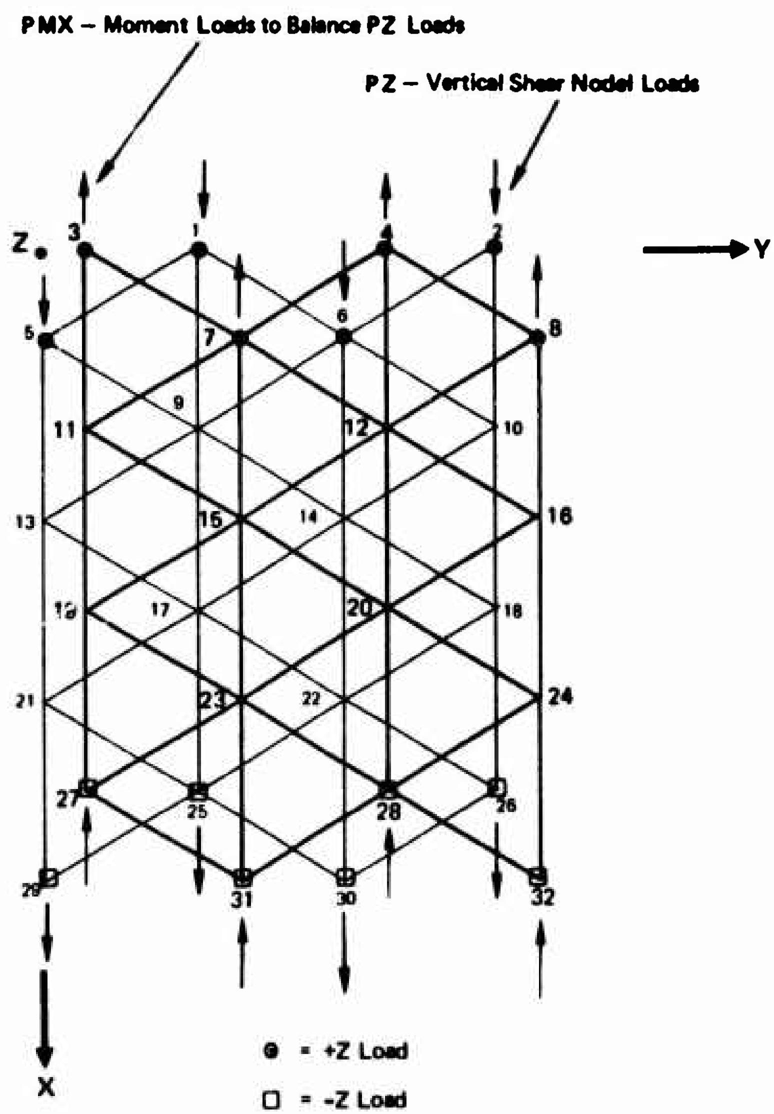


Figure 44. Vertical Shear Load XOSHR Applied to Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

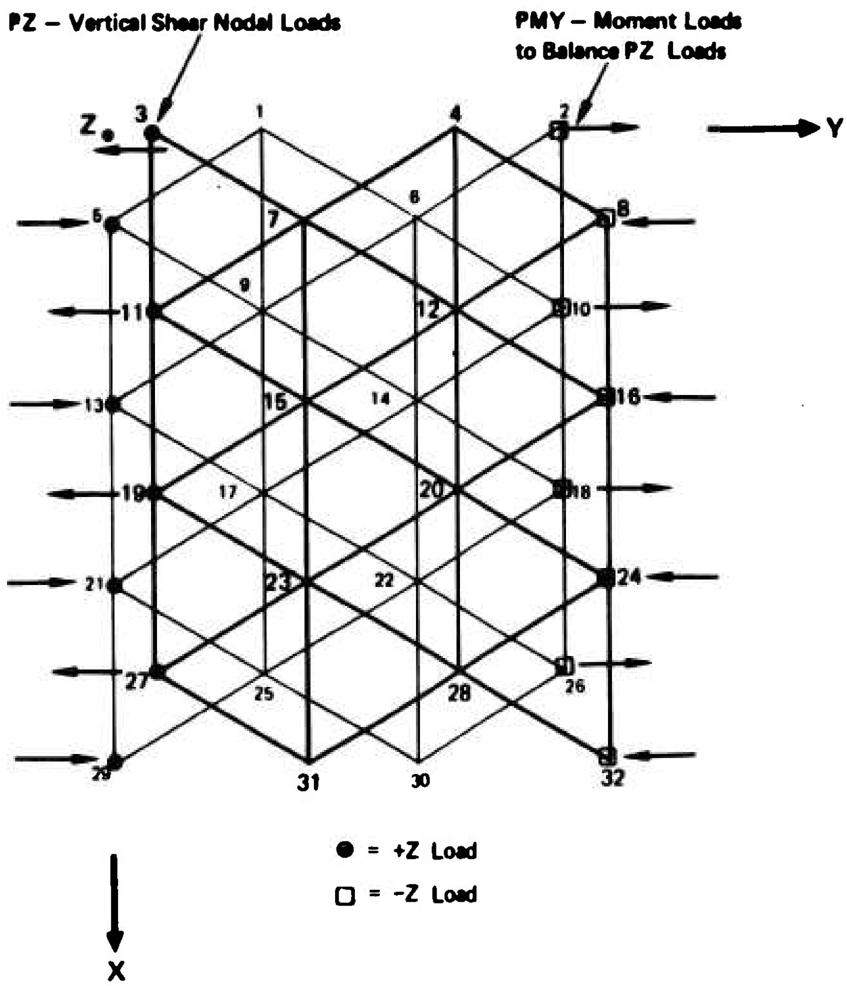


Figure 45. Vertical Shear Load YQSHR Applied to Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

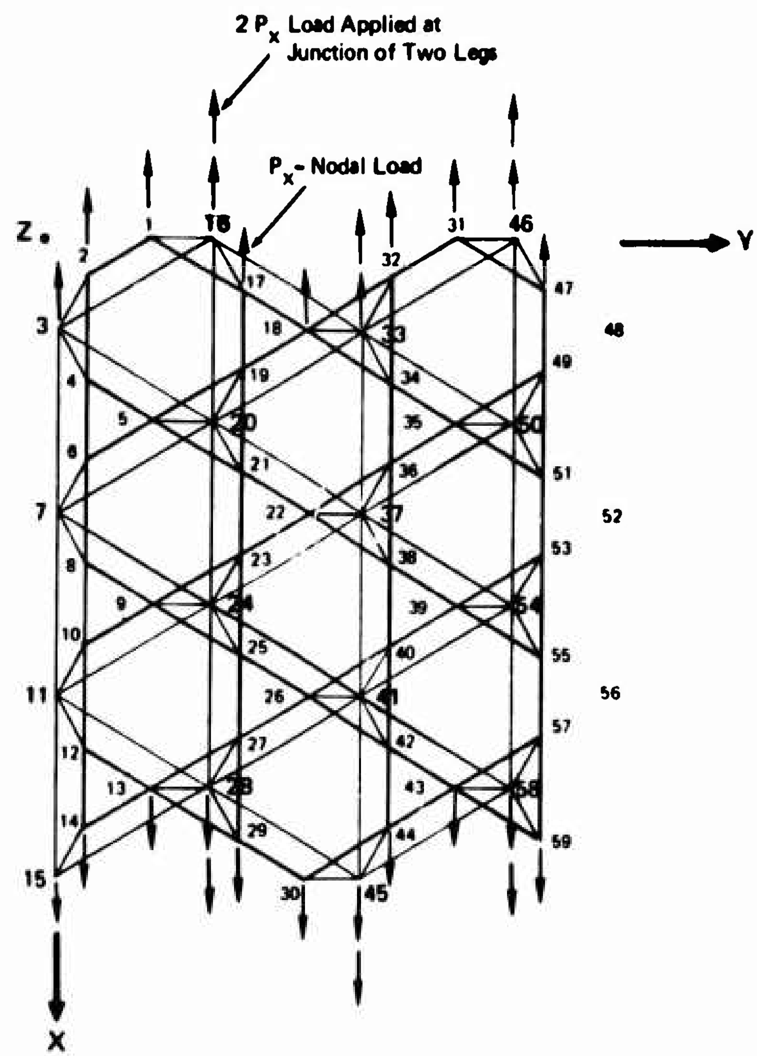


Figure 46. Load N_X Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

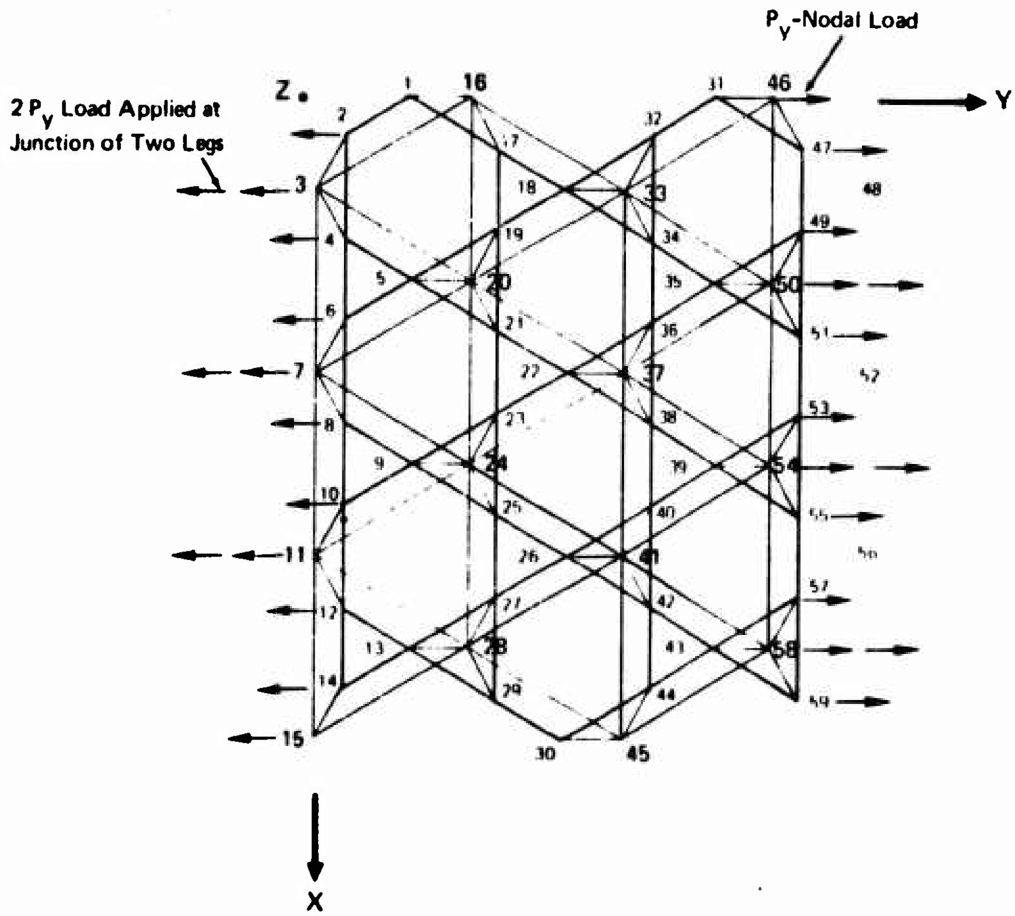


Figure 47. **Load N_Y Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.**

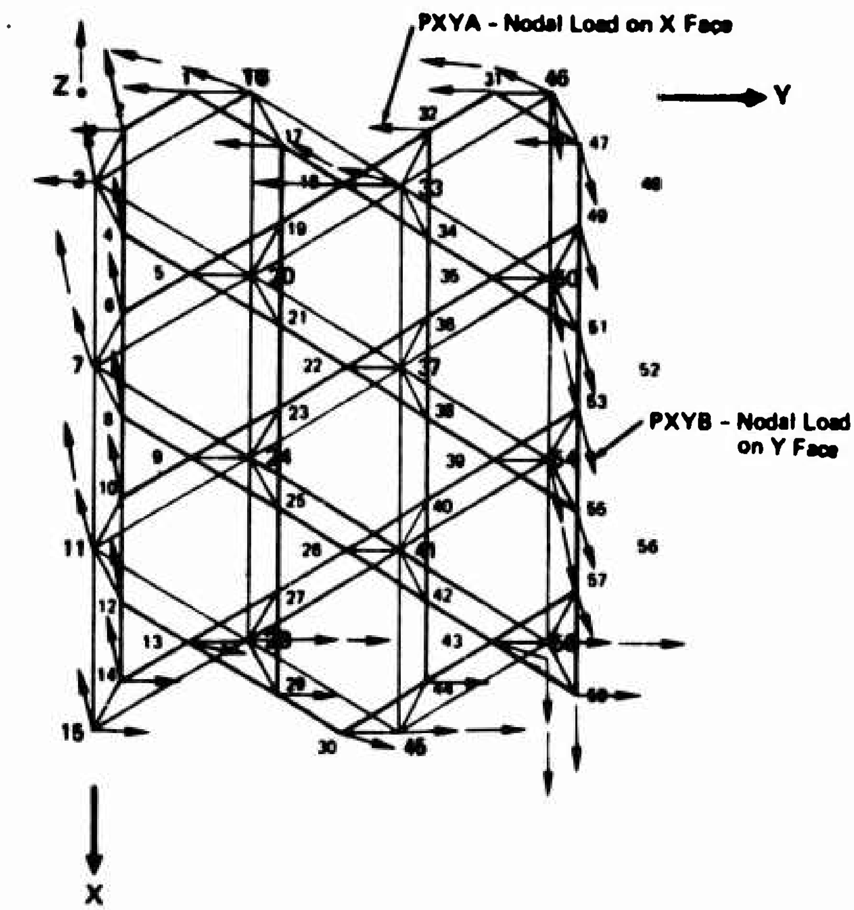


Figure 48. In-Plane Shear Load N_{XY} Applied to Truncated Tetra-Core Flat Plate, $L_X = 4, L_Y = 4.$

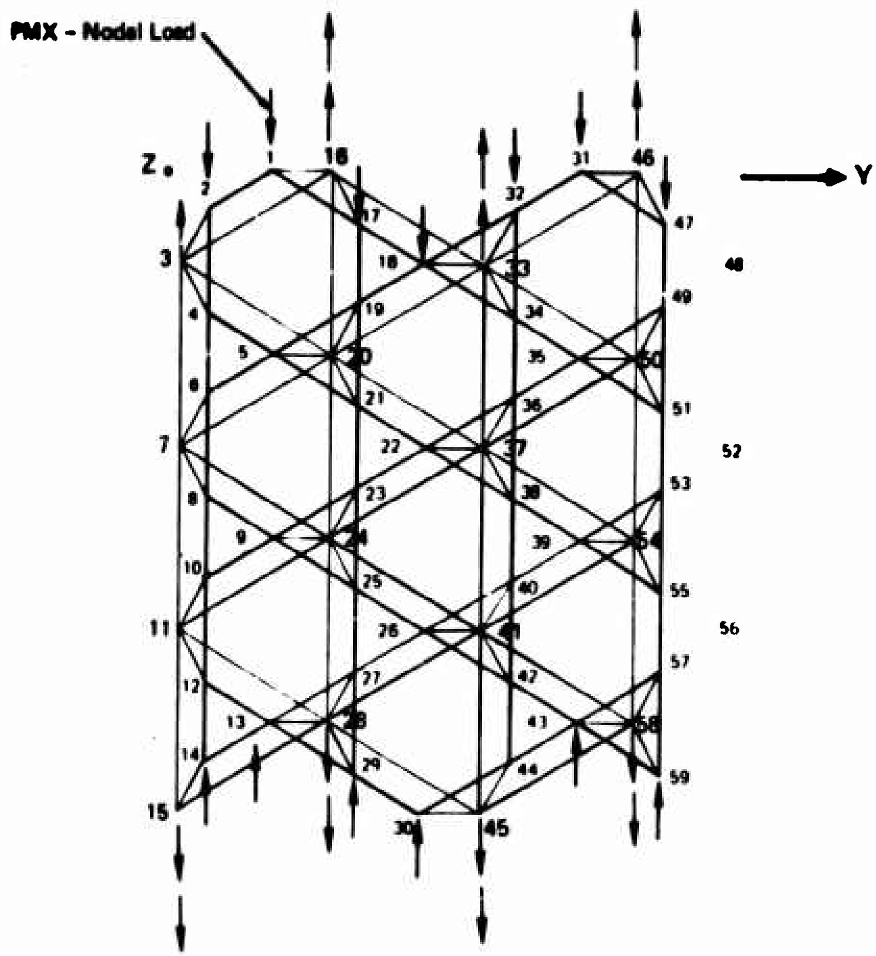


Figure 49. Moment M_X Applied to Truncated Tetra-Core Flat Plate, L_X = 4, L_Y = 4.

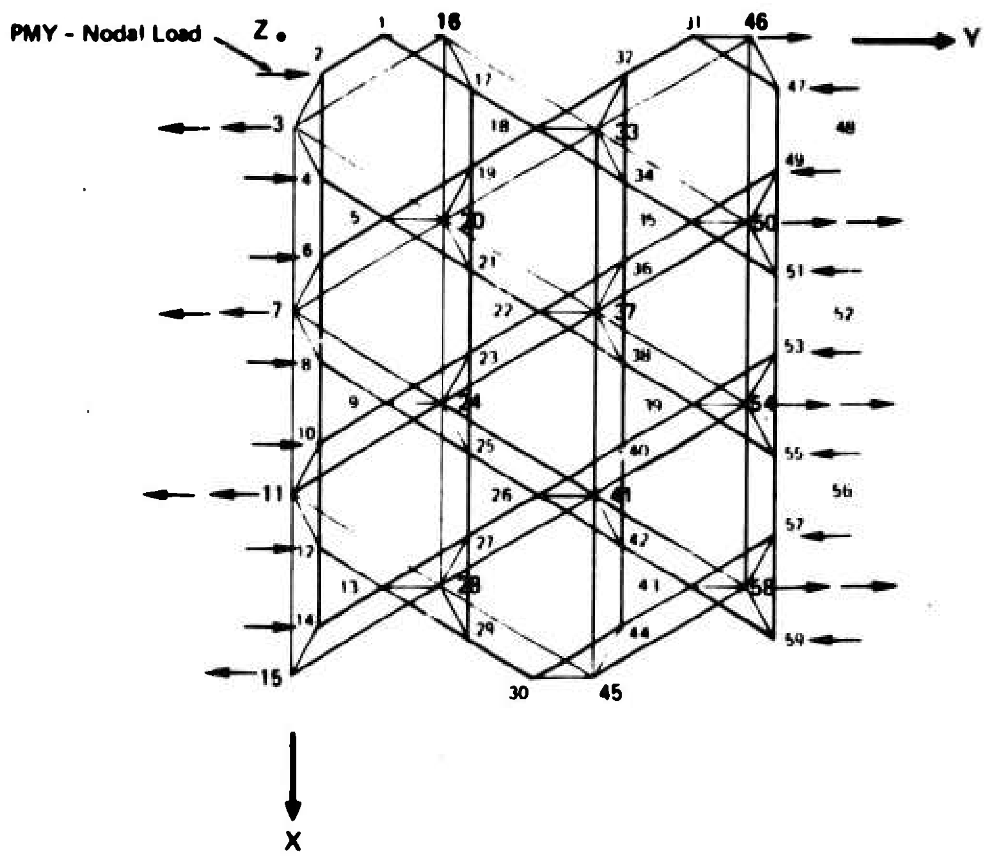


Figure 50. Moment M_Y Applied to Truncated Tetra-Core Flat Plate, $L_X = 4$, $L_Y = 4$.

APPENDIX II
EXAMPLE OF COMPUTER PROGRAM OUTPUTS

An example of the output from an optimization run is given (Figure 51). Stresses and deflections for the input configuration are printed. Then the optimization steps are printed. Finally, stresses and deflections for the optimized section are printed. Element stresses are printed out of sequence in this example, although they are in sequence in the latest version of the program.

```
LOADS      OPT      PLASTIC      TYPE      FACE      HOLE      STEPS      TETRA
  1          1         0           1         0         0          3
TETRA-CORE FLAT PLATE ANALYSIS

OPTIMIZATION OPTIONS
ISA      ISE      ISC
 12       -0       -0
 -0       -0       -0
 -0       -0       -0
  5       -0       -0
  4       -0       -0
 12       -0       -0
 12       -0       -0
 12       -0       -0
 12       -0       -0
 12       -0       -0
 12       -0       -0

TETRA-CORE PLATE STRESSES

EX          EY          GXY         MUXY
4000000     5000000     500000      .25000
4000000     5000000     500000      .25000
4000000     5000000     500000      .25000
   LX          LY     SIDE      HEIGHT          LEG THICKNESSES
   4           4      .75000    .50000     .03000      .03000      .03000

THETA       Y1OFF       Y2OFF       THT
-0.0000     -0.0000     -0.0000     .5000

     ALLOWABLE STRESSES
```

Figure 51. Output from Optimization Run.

FXT	FYT	FXC	FYC	FXY
110000	13600	110000	13600	11000
110000	13600	110000	13600	11000
110000	13600	110000	13600	11000

LEG DENSITY = .10000 .10000 .10000

FIRST VERTICAL PLATE = 1 SKEW A = 25 SKEW B = 46 LWR FACE = 0 UPPER FACE = 0

LOAD CASE 1

NX	NY	NXY	MX	MY	MXY	TORQUE	GX	GY
1000	-0	100	-0	-0	-0	-0	-0	-0

```
NUMBER OF NODAL POINTS------- 32
NUMBER OF ELEMENTS----------- 66
NUMBER OF LOAD CASES      --- 1
```

NODAL POINT	X	Y	Z
1	0.0000	.0499	0.0000
2	0.0000	1.9498	0.0000
3	0.0000	.2165	.5000
4	0.0000	1.5164	.5000
5	.3750	0.0000	0.0000
6	.3750	1.2999	0.0000
7	.3750	.8664	.5000
8	.3750	2.1663	.5000
9	.7500	.0493	0.0000
10	.7500	1.9498	0.0000
11	.7500	.2165	.5000
12	.7500	1.5164	.5000
13	1.1250	0.0000	0.0000
14	1.1250	1.2999	0.0000
15	1.1250	.8664	.5000
16	1.1250	2.1663	.5000
17	1.5000	.0493	0.0000
18	1.5000	1.9498	0.0000
19	1.5000	.2165	.5000
20	1.5000	1.5164	.5000
21	1.8750	0.0000	0.0000
22	1.8750	1.2999	0.0000
23	1.8750	.8664	.5000
24	1.8750	2.1663	.5000
25	2.2500	.0493	0.0000
26	2.2500	1.9498	0.0000
27	2.2500	.2165	.5000
28	2.2500	1.5164	.5000
29	2.6250	0.0000	0.0000
30	2.6250	1.2999	0.0000
31	2.6250	.8664	.5000
32	2.6250	2.1663	.5000

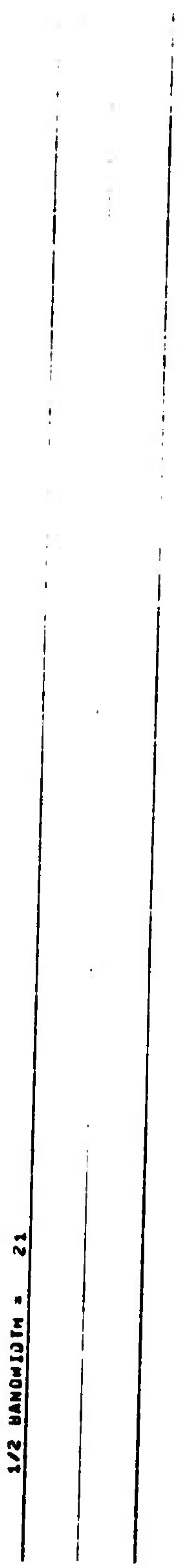
LOAD CASE 1

NEW NODE	OLD NODE	UX	UY	UZ
1	1	-60.1746	-6.0175	0.0000
2	2	-52.8629	-6.0175	0.0000
3	3	-67.4662	-6.0175	0.0000
4	3001	-209.8693	-24.0693	0.0000
5	4	-60.1746	-6.0175	0.0000
6	4002	-240.6963	-24.0693	0.0000
7	5	-67.4662	-6.0175	0.0000
8	5003	-209.8693	-24.0693	0.0000
9	6	-60.1746	-6.0175	0.0000
10	6004	-240.6963	-24.0693	0.0000
11	7	-60.1746	-6.0175	0.0000
12	7001	-240.6963	-24.0693	0.0000
13	7006	-240.6963	-24.0693	0.0000
14	8	-52.8825	-6.0175	0.0000
15	8002	-211.5317	-24.0693	0.0000
16	9	0.0000	0.0000	0.0000
17	9007	0.0000	0.0000	0.0000
18	10	7.2917	0.0000	0.0000
19	10008	0.0000	0.0000	0.0000
20	11	-7.2917	0.0000	0.0000
21	11009	-209.8693	0.0000	0.0000
22	11029	0.0000	0.0000	0.0000
23	12	0.0000	0.0000	0.0000
24	12006	0.0000	0.0000	0.0000
25	12010	0.0000	0.0000	0.0000
26	13	-7.2917	0.0000	0.0000
27	13011	-29.1667	0.0000	0.0000
28	14	0.0000	0.0000	0.0000
29	14012	0.0000	0.0000	0.0000
30	15	0.0000	0.0000	0.0000
31	15009	0.0000	0.0000	0.0000
32	15014	0.0000	0.0000	0.0000
33	16	7.2917	0.0000	0.0000
34	16010	29.1667	0.0000	0.0000
35	17	0.0000	0.0000	0.0000
36	17015	0.0000	0.0000	0.0000
37	18	7.2917	0.0000	0.0000
38	18016	29.1667	0.0000	0.0000
39	19	-7.2917	0.0000	0.0000
40	19013	-29.1667	0.0000	0.0000
41	19017	0.0000	0.0000	0.0000
42	20	0.0000	0.0000	0.0000
43	20014	0.0000	0.0000	0.0000
44	20018	0.0000	0.0000	0.0000
45	21	-7.2917	0.0000	0.0000
46	21019	-29.1667	0.0000	0.0000
47	22	0.0000	0.0000	0.0000
48	22020	0.0000	0.0000	0.0000
49	23	0.0000	0.0000	0.0000
50	23017	0.0000	0.0000	0.0000
51	23022	0.0000	0.0000	0.0000
52	24	7.2917	0.0000	0.0000

53	24018	29.1667	0.0000	0.0000
54	25	60.1746	6.0175	0.0000
55	25023	0.0000	0.0000	0.0000
56	26	67.4062	6.0175	0.0000
57	26024	29.1667	0.5000	0.0000
58	27	32.3629	6.0175	0.0000
59	27021	0.0000	0.0000	0.0000
60	27025	211.5317	24.0699	0.0000
61	28	60.1746	6.0175	0.0000
62	28022	0.0000	0.0000	0.0000
63	28026	240.6983	24.0699	0.0000
64	29	32.3629	6.0175	0.0000
65	29027	211.5317	24.0699	0.0000
66	30	60.1746	6.0175	0.0000
67	30028	240.6983	24.0699	0.0000
68	31	60.1746	6.0175	0.0000
69	31025	240.6983	24.0699	0.0000
70	31030	240.6983	24.0699	0.0000
71	32	67.4062	6.0175	0.0000
72	32026	269.8650	24.0699	0.0000

ELEMENT NUMBER	I	J	K	EX	EY	EXY	G	T	LF	LQ	LR	LS	LT
1	3	11	5	4031496	503937	125984	500000	.0300	3	20	7	21	6
2	5	13	11	4031496	503937	125984	500000	.0300	7	26	20	27	21
3	11	19	13	4031496	503937	125984	500000	.0300	20	39	26	40	27
4	13	21	19	4031496	503937	125984	500000	.0300	26	45	39	46	40
5	19	27	21	4031496	503937	125984	500000	.0300	39	58	45	59	46
6	21	29	27	4031496	503937	125984	500000	.0300	45	64	58	65	59
7	1	9	7	4031496	503937	125984	500000	.0300	1	16	11	17	12
8	7	15	9	4031496	503937	125984	500000	.0300	11	30	16	31	17
9	9	17	15	4031496	503937	125984	500000	.0300	16	35	30	36	31
10	15	23	17	4031496	503937	125984	500000	.0300	30	49	35	50	36
11	17	25	23	4031496	503937	125984	500000	.0300	35	54	49	55	50
12	23	31	25	4031496	503937	125984	500000	.0300	49	68	54	69	55
13	4	12	6	4031496	503937	125984	500000	.0300	5	23	9	24	10
14	6	14	12	4031496	503937	125984	500000	.0300	9	28	23	29	24
15	12	20	14	4031496	503937	125984	500000	.0300	23	42	28	43	29
16	14	22	20	4031496	503937	125984	500000	.0300	28	47	42	48	43
17	20	28	22	4031496	503937	125984	500000	.0300	42	61	47	62	48
18	22	30	28	4031496	503937	125984	500000	.0300	47	66	61	67	62
19	2	10	8	4031496	503937	125984	500000	.0300	2	18	14	19	15
20	8	16	10	4031496	503937	125984	500000	.0300	14	33	18	34	19
21	10	18	16	4031496	503937	125984	500000	.0300	18	37	33	38	34
22	16	24	18	4031496	503937	125984	500000	.0300	33	52	37	53	38
23	18	26	24	4031496	503937	125984	500000	.0300	37	56	52	57	53
24	24	32	26	4031496	503937	125984	500000	.0300	52	71	56	72	57
25	1	5	3	4031496	503937	125984	500000	.0300	1	7	3	8	4
26	2	6	4	4031496	503937	125984	500000	.0300	2	9	5	10	6
27	4	7	6	4031496	503937	125984	500000	.0300	5	11	9	13	10
28	6	9	7	4031496	503937	125984	500000	.0300	9	16	11	17	13
29	7	11	9	4031496	503937	125984	500000	.0300	11	20	16	22	17
30	9	13	11	4031496	503937	125984	500000	.0300	16	26	20	27	22
31	8	12	10	4031496	503937	125984	500000	.0300	14	23	18	25	19
32	10	14	12	4031496	503937	125984	500000	.0300	18	28	23	29	25
33	12	15	14	4031496	503937	125984	500000	.0300	23	30	28	32	29
34	14	17	15	4031496	503937	125984	500000	.0300	28	35	30	36	32
35	15	19	17	4031496	503937	125984	500000	.0300	30	39	35	41	36
36	17	21	19	4031496	503937	125984	500000	.0300	35	45	39	46	41
37	16	20	18	4031496	503937	125984	500000	.0300	33	42	37	44	38
38	18	22	20	4031496	503937	125984	500000	.0300	37	47	42	48	44
39	20	23	22	4031496	503937	125984	500000	.0300	42	49	47	51	48
40	22	25	23	4031496	503937	125984	500000	.0300	47	54	49	55	51
41	23	27	25	4031496	503937	125984	500000	.0300	49	58	54	60	55
42	25	29	27	4031496	503937	125984	500000	.0300	54	64	58	65	60
43	24	28	26	4031496	503937	125984	500000	.0300	52	61	56	63	57
44	26	30	28	4031496	503937	125984	500000	.0300	56	66	61	67	63
45	28	31	30	4031496	503937	125984	500000	.0300	61	68	66	70	67
46	4	8	2	4031496	503937	125984	500000	.0300	5	14	2	15	6
47	3	7	1	4031496	503937	125984	500000	.0300	3	11	1	12	4
48	1	6	7	4031496	503937	125984	500000	.0300	1	9	11	13	12
49	7	12	6	4031496	503937	125984	500000	.0300	11	23	9	24	13
50	6	10	12	4031496	503937	125984	500000	.0300	9	18	23	25	24

ELEMENT NUMBER	I	J	K	EX	EY	EXY	G	T	LF	LU	LR	LS	LT
51	12	16	10	4031496	503937	125984	500000	0.300	25	33	16	34	25
52	5	9	11	4031496	503937	125984	500000	0.300	7	10	20	22	21
53	11	15	9	4031496	503937	125984	500000	0.300	20	30	18	31	22
54	9	14	15	4031496	503937	125984	500000	0.300	10	25	30	32	31
55	15	20	14	4031496	503937	125984	500000	0.300	30	42	22	43	32
56	14	18	20	4031496	503937	125984	500000	0.300	28	37	42	44	43
57	20	24	18	4031496	503937	125984	500000	0.300	42	52	37	53	44
58	13	17	19	4031496	503937	125984	500000	0.300	26	35	39	41	40
59	19	23	17	4031496	503937	125984	500000	0.300	23	49	35	50	41
60	17	22	23	4031496	503937	125984	500000	0.300	35	47	49	51	50
61	23	28	22	4031496	503937	125984	500000	0.300	49	61	47	62	51
62	22	26	28	4031496	503937	125984	500000	0.300	47	58	61	63	62
63	28	32	26	4031496	503937	125984	500000	0.300	61	71	56	72	63
64	21	25	27	4031496	503937	125984	500000	0.300	45	54	56	60	59
65	27	31	25	4031496	503937	125984	500000	0.300	58	69	54	65	64
66	25	30	31	4031496	503937	125984	500000	0.300	54	66	65	70	69


1/2 BANDWIDTH = 21

DEFLECTIONS LOAD CASE 1

NODAL POINT	UX	UY	UZ
1	-1.1062982E-02	1.2413646E-02	-1.8444397E-02
2	4.7463039E-03	4.4597166E-03	-1.7565324E-02
3	-2.9341568E-02	1.1782339E-02	-7.3454502E-03
3001	-3.3944329E-02	1.3387172E-02	-1.2056579E-02
4	-9.9550079E-03	3.5473393E-04	-1.1794306E-02
4002	-1.4786092E-02	9.4919973E-04	-1.5476075E-02
5	-2.3164284E-02	1.1919735E-02	1.7729434E-04
5003	-2.5235001E-02	1.0965123E-02	-3.9923594E-03
6	-1.5310636E-03	3.4940571E-03	-9.2730449E-03
6004	-4.3063569E-03	5.5591366E-03	-1.0132757E-02
7	-1.4053242E-02	1.4499913E-03	-1.1508398E-02
7001	-9.6026975E-03	1.3099453E-03	-1.5412439E-02
7006	-1.0408842E-02	2.2789610E-03	-1.0032635E-02
8	-5.3714118E-04	-5.5496662E-03	-1.5356486E-02
8002	4.8900670E-03	-9.2643846E-03	-1.5559486E-02
9	-3.1922383E-15	1.5729257E-15	-1.5481304E-15
9007	-7.0524317E-03	3.3842879E-03	-6.8466256E-03
10	1.1485845E-02	-3.1050342E-03	-6.5118781E-03
10008	7.0480530E-03	-1.0476439E-03	-1.0913726E-02
11	-1.9879627E-02	3.1252393E-03	6.0531576E-04
11005	-2.2841638E-02	9.0458042E-03	1.4244682E-04
11009	-1.3087123E-02	1.9470746E-03	6.6255581E-07
12	-2.4042270E-03	-3.4569179E-03	-5.4744508E-03
12006	-1.5947318E-02	-1.1206104E-03	-6.8289555E-03
12018	3.3723237E-03	-3.1526941E-03	-5.8846579E-03
13	-1.5174995E-02	2.5306638E-04	2.6174545E-03
13011	-1.8718359E-02	-2.7227034E-03	2.9803995E-03
14	5.2220435E-03	-2.4943692E-03	8.1528775E-27
14012	1.7170295E-03	-2.8041488E-03	-2.5835534E-03
15	-4.7623288E-03	-2.6591416E-03	1.8356457E-04
15009	-3.8193293E-03	-9.6379471E-04	1.9265216E-04
15014	-1.5522870E-04	-2.5768343E-03	2.2392411E-04
16	7.7012324E-03	-8.2066538E-03	-7.3987366E-04
16018	9.9471793E-03	-6.7161635E-03	-3.2867575E-03
17	1.1853389E-03	-1.5729257E-15	1.5481304E-15
17015	-9.6898834E-04	-1.4338798E-03	1.3675527E-03
18	1.6775446E-02	-6.6134611E-03	4.3436681E-03
18016	1.2952588E-02	-5.1666942E-03	1.0431157E-03
19	-1.2570513E-02	-3.7703652E-03	7.5414567E-04
19013	-1.4987469E-02	2.7014679E-03	3.3906424E-04
19017	-5.4967213E-03	-2.4256833E-03	5.7171534E-04
20	4.9083268E-03	-3.6554557E-03	3.2146352E-03
20014	5.1460646E-03	-5.4163835E-03	1.4094193E-03
20018	9.6143110E-03	-7.6479233E-03	3.6943215E-03
21	-8.7798078E-03	-3.9815035E-04	1.3870734E-03
21019	-1.1560545E-02	-5.1322766E-03	2.0726992E-03
22	1.1928084E-02	-6.0163788E-03	7.3727903E-03
22020	8.0568243E-03	-7.3664301E-03	5.5294590E-03
23	1.5781611E-03	-1.0124793E-02	5.1352571E-03
23017	1.7563289E-03	-5.3626746E-03	3.0815556E-03

DEFLECTIONS LOAD CASE 1

NODAL POINT	UX	UY	UZ
23022	6.8426785E-03	-8.3568100E-03	6.2433531E-03
24	1.4950745E-02	-1.7388066E-02	1.3501318E-02
24018	1.6023437E-02	-1.1530498E-02	7.2378430E-03
25	9.3867077E-03	-6.4083651E-03	9.0155342E-03
25023	4.7572811E-03	-9.1140575E-03	6.8560563E-03
26	2.3811766E-02	-1.2835119E-02	1.5276522E-02
26024	1.8947976E-02	-1.7035742E-02	1.3122416E-02
27	-5.7743278E-03	-1.1442523E-02	3.7966276E-03
27021	-6.5317552E-03	-1.4849790E-03	2.1452081E-03
27025	5.7949423E-03	-6.7352359E-03	6.1717381E-03
28	1.4246730E-02	-1.7691984E-02	1.0303360E-02
28022	1.2106550E-02	-1.3755932E-02	9.9512777E-03
28026	2.2795811E-02	-1.5409312E-02	1.2980581E-02
29	-6.3238010E-03	-1.0714146E-02	5.3756521E-03
29027	-6.2461130E-03	-1.4971960E-02	4.7206483E-03
30	1.9066498E-02	-1.4525970E-02	1.2687533E-02
30028	1.4868791E-02	-1.3792661E-02	1.3879110E-02
31	8.3256107E-03	-1.0076274E-02	1.5773595E-02
31025	7.2064386E-03	-9.0741050E-03	1.2727512E-02
31030	2.3137753E-02	-1.6533126E-02	1.3962538E-02
32	1.9550315E-02	-1.9342500E-02	2.3274549E-02
32026	2.1351489E-02	-1.4722069E-02	1.8905125E-02

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
1	3	11	5	48462.652	-3815.282	-5220.402	.439	1
				51758.415	950.891	1806.545		
				377.557	-860.648	2727.222		
				33532.941	-1231.043	-223.512		
2	5	13	11	30561.693	319.136	-6512.023	.468	1
				54459.352	-1559.038	-5363.536		
				45347.761	-109.294	28.759		
				43450.334	-243.352	-3554.317		
3	11	19	13	38756.365	1676.545	-2315.514	.910	1
				39128.773	-2189.409	-623.487		
				40495.441	-459.694	4101.706		
				39443.703	-324.206	367.536		
4	13	21	19	25563.149	1209.331	-5976.503	.679	1
				42427.236	-2343.430	-1838.154		
				35781.015	-1396.465	-539.134		
				35523.967	-822.355	-2702.547		
7	1	9	7	69392.392	-5300.132	2415.041	.289	1
				47569.031	3313.259	6648.442		
				-4400.761	704.477	1309.355		
				37553.714	-596.132	3077.530		
8	7	15	9	52560.744	191.111	-3412.763	.769	1
				46711.376	-661.558	2825.313		
				18571.323	-3826.562	4766.527		
				39114.465	-1432.353	1393.217		
9	9	17	15	-5659.279	-1642.188	-6622.504	.231	1
				18138.180	1199.329	-3457.448		
				54151.727	-1334.962	-2164.282		
				22211.542	-559.274	-4061.598		
10	15	23	17	46512.183	-1344.438	3431.444	1.009	1
				26616.663	576.521	2009.517		
				24842.290	2459.267	1324.509		
				30790.173	416.430	2455.177		
13	4	12	6	49954.496	-3434.141	2776.347	.764	1
				50313.463	2369.114	4303.573		
				17651.469	122.796	2560.151		
				32623.125	-324.077	3401.990		
14	6	14	12	37243.774	-1588.662	-1146.566	1.575	1
				34524.931	530.914	2515.133		
				34555.645	-1074.767	-75.112		
				35378.117	-713.839	430.565		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
15	12	20	14	39744.865	-594.133	-331.423	1.642	1
				38581.017	85.321	1473.297		
				33437.964	-911.328	-408.937		
				37408.222	-473.382	77.756		
16	14	22	20	36863.440	-388.636	-473.592	1.200	1
				34672.587	168.281	1411.304		
				38870.852	-1039.216	148.035		
				38782.288	-415.837	359.036		
19	2	10	8	42973.272	-8372.855	-378.373	.080	1
				28018.573	4785.670	3370.111		
				9812.592	-1920.892	-1272.401		
				26868.140	-1835.759	2272.413		
20	8	10	10	47388.249	-2001.449	797.500	1.016	1
				40216.145	1553.927	3554.253		
				17825.126	-339.634	-1319.552		
				35141.163	-483.385	1012.234		
21	13	16	16	30694.539	-791.489	1443.329	1.566	1
				25750.544	882.414	1532.358		
				35853.581	-201.501	-372.753		
				30766.313	-36.859	701.179		
25	1	5	3	-44477.740	14567.516	-6170.169	-.043	2
				-24291.442	-23100.355	-16941.290		
				56871.919	788.461	12151.695		
				-632.421	1756.539	-3521.515		
26	2	6	4	-23517.348	11173.719	-1259.235	-.764	2
				147.263	-6693.437	-13273.742		
				39332.123	1355.551	4635.422		
				5297.345	2156.944	-3298.302		
27	4	7	6	-22618.079	3493.625	-4265.574	-.692	2
				-9452.561	-3386.978	-3331.369		
				12264.771	427.473	2583.528		
				-6621.963	-21.960	-1837.545		
28	6	9	7	7687.344	3651.352	-7369.551	.226	2
				32614.975	-4148.254	-4647.544		
				-5679.000	-984.407	-615.523		
				11541.084	-451.123	-4091.530		
29	7	11	9	-28280.947	2765.071	-3745.416	-.754	2
				-17837.985	-1345.242	-3186.436		
				16851.066	-2447.335	2363.477		
				-9775.949	-362.493	-1357.449		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
30	9	13	11	-38145.532	-4671.025	2558.538	-.045	2
				-44645.083	5567.258	1061.453		
				35512.575	-252.672	-565.234		
				-15726.162	214.364	1313.572		
31	8	12	10	-24645.521	2721.341	-3375.525	-.726	2
				-4314.724	-2259.411	-5631.535		
				3255.114	50.722	-1353.703		
				-8621.511	166.384	-3463.578		
32	10	14	12	-19224.070	1153.215	793.539	-.644	2
				-19549.565	-571.924	-1259.516		
				-3365.907	646.777	861.561		
				-14665.114	463.350	131.537		
33	12	15	14	-10656.370	378.169	-168.106	-.565	2
				-7331.552	-336.473	-1127.540		
				-16653.126	634.538	-611.534		
				-11354.752	225.391	-635.764		
34	14	17	15	-25531.077	155.844	1613.290	-.725	2
				-18710.263	1784.362	-4116.534		
				-7214.767	1163.950	-1351.235		
				-17667.565	1356.255	-1164.564		
35	15	19	17	-14773.554	-915.585	4304.196	-.622	2
				-18435.455	411.183	-1537.538		
				-12575.643	5255.531	-1128.758		
				-15654.964	1585.143	465.573		
46	4	8	2	-21652.635	15127.662	-11678.536	-.676	3
				17127.654	-13666.536	-15158.416		
				23271.555	[illegible]	12457.539		
				6435.055	563.368	-4700.525		
47	3	7	1	-18123.755	12567.533	-2120.155	-.616	3
				5310.157	-3351.465	-14563.126		
				43553.525	2155.445	5028.556		
				10348.055	2247.664	-2691.442		
48	1	6	7	-28437.513	4212.745	-6306.716	-.755	3
				-3341.477	-4556.226	-5685.542		
				24448.553	-465.203	2566.537		
				-2443.255	-416.260	-3762.522		
49	7	12	6	440.169	2150.554	-5348.133	.556	3
				16674.416	-2356.566	-2459.528		
				22725.565	-1325.739	213.507		
				13146.625	-665.057	-2531.296		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
50	6	10	12	-2611.643	965.563	-3566.499	1.476	3
				11166.753	-4235.420	-2937.495		
				11353.564	179.519	-935.411		
				6569.031	-23.339	-2359.735		
51	12	10	10	1612.734	1056.024	-769.341	2.936	3
				8224.763	-1668.547	-2358.439		
				-443.339	1166.562	35.320		
				3181.355	374.846	-1333.320		
52	5	9	11	10634.673	-2542.410	-755.135	1.292	3
				1735.564	-666.452	4705.536		
				-1676.573	533.026	559.534		
				3655.560	-795.513	1553.445		
53	11	15	7	16361.260	665.294	-1377.536	1.262	3
				9165.675	750.982	3254.412		
				-1335.275	-5261.764	1902.547		
				6663.864	-1290.109	1372.473		
54	9	14	15	-5476.143	-2263.667	-6112.532	.271	3
				9711.651	-41.556	-562.556		
				7136.706	-736.731	-1110.555		
				3790.663	-967.456	-2573.536		
55	15	20	14	-6524.973	-[illegible]	612.797	1.655	3
				-2438.369	246.154	21.252		
				5745.116	362.566	250.456		
				617.756	125.553	278.532		
56	14	18	22	6264.254	474.511	-575.517	-.121	3
				12721.420	-749.645	-973.655		
				-7916.674	705.124	-456.703		
				4563.060	143.463	-565.553		
58	13	17	19	45609.667	-5210.566	25536.54	-.611	2
				39632.715	5166.172	3444.536		
				-36765.365	1365.536	-1361.593		
				16225.575	416.631	593.542		
[illegible]	19	27	23	30415.965	2271.651	-4553.426	.965	1
				41752.964	-3457.553	-1166.517		
				27665.665	-[illegible]	-244.731		
				33651.539	-[illegible]	-1655.448		
[illegible]	21	29	27	6425.466	[illegible]	-3255.[illegible]	1.636	1
				15919.565	-[illegible]	-2455.514		
				35276.766	-1551.551	-1263.21		
				26545.596	-256.374	-2515.514		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LCD
11	17	25	23	42990.800	1274.415	-309.202	1.455	1
				44544.243	-1327.312	3.713		
				20130.174	-599.025	-604.012		
				35891.072	-116.974	-323.720		
12	23	31	25	27012.022	2490.353	-470.050	.764	1
				44520.479	-4757.553	-2524.037		
				16516.544	1317.523	-1792.144		
				29349.068	-247.080	-3025.497		
17	20	28	22	47187.610	410.672	-2553.455	[illegible]	1
				51314.576	-1426.292	432.35		
				24054.530	-645.544	164.359		
				40852.308	-553.321	-645.352		
18	22	30	28	27401.330	3274.974	-4935.730	.573	1
				48202.014	-4935.062	-2900.592		
				20970.646	330.285	-1713.447		
				32211.793	-442.600	-3533.797		
22	16	24	18	42303.942	-594.539	1315.424	1.461	1
				34909.401	233.645	2389.736		
				26825.087	-80.335	-137.023		
				34679.477	-163.007	1172.996		
23	18	26	24	36384.069	699.495	-1325.037	1.707	1
				38479.560	-1402.144	332.540		
				24738.243	-499.046	-927.00		
				33200.023	-420.565	-441.706		
24	24	32	26	15335.110	1137.953	-1033.074	1.026	1
				23101.475	-651.590	-1381.000		
				31958.183	-477.124	-2377.021		
				23491.591	3.091	-1497.005		
36	17	21	19	-10848.363	-534.449	9435.932	-.212	2
				-37205.273	3412.089	3132.077		
				10738.571	333.930	758.726		
				-12258.350	1053.873	4443.752		
37	16	20	18	-7973.100	2193.630	-772.040	-.127	2
				-225.045	-1352.590	-539.045		
				-212.840	425.820	175.037		
				-2863.930	421.024	-1230.753		
38	18	22	20	-14119.033	533.970	230.044	-.507	2
				-13430.021	-126.077	-307.035		
				4067.305	266.142	595.034		
				-7827.514	223.814	-495.035		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
39	20	23	22	-3390.224	-334.477	507.472	.008	2
				-3023.372	473.934	-529.431		
				-6904.575	567.238	-322.330		
				-4439.390	242.232	-94.339		
40	22	25	23	-5630.967	-271.830	1439.377	-.244	2
				-9205.011	549.454	77.025		
				-256.453	216.297	229.344		
				-5164.137	297.973	632.162		
41	23	27	25	-4112.574	-2410.394	4253.036	-.291	2
				-22556.049	3769.302	3530.459		
				17162.770	-1404.926	1704.031		
				-3162.151	-43.987	3172.035		
42	25	29	27	11672.010	-4311.575	2502.002	-.620	2
				-1050.001	3130.740	3503.135		
				-18322.730	1034.959	236.932		
				-2569.567	-45.956	2360.473		
43	24	28	26	11244.051	-3056.739	5138.709	-.328	2
				-10356.364	3021.157	6033.546		
				11500.515	-558.493	-1372.314		
				4129.394	-211.359	3300.030		
44	26	30	28	5670.706	-3750.019	5863.507	-.557	2
				-15715.335	3383.567	6314.595		
				296.091	367.172	-2506.931		
				-3182.809	-38.027	3196.096		
45	28	31	30	-.633	-5801.034	11345.055	-.676	2
				-21667.390	3209.608	2713.064		
				39436.126	1154.819	-5390.332		
				5942.092	1520.751	2868.452		
57	20	24	18	-12768.574	706.271	649.073	-.458	3
				-12856.881	-334.820	-304.344		
				13187.823	175.575	-303.734		
				-4145.812	163.039	-26.258		
59	19	23	17	9730.580	113.710	1114.705	1.768	3
				6018.515	-1546.330	-1242.079		
				4578.323	3480.381	2451.045		
				6575.860	869.254	774.353		
60	17	22	23	-3708.721	1989.040	504.777	.277	3
				178.234	-603.560	-1851.058		
				11825.748	-262.935	-392.030		
				2765.087	367.533	-720.226		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
61	23	20	22	2052.063	-731.834	2046.548	.122	3
				-6203.757	1051.017	1013.422		
				13561.112	-153.462	535.504		
				3250.579	55.241	1365.545		
62	22	26	28	12176.506	-3391.090	3500.251	.132	3
				-6149.215	3792.520	4487.120		
				23568.475	-1552.223	876.731		
				9865.256	-283.795	3054.713		
63	28	32	26	14583.083	-3926.403	2364.514	-.348	3
				3933.226	2912.590	2333.493		
				-10640.550	1403.063	-1361.578		
				2625.463	156.397	1667.443		
64	21	25	27	36735.575	-5767.677	7056.541	.037	3
				3709.674	4056.321	3927.524		
				-1683.626	-318.575	-1494.644		
				12920.541	-499.477	5163.540		
65	27	31	25	13763.401	-3780.866	5815.427	-.062	3
				-7419.356	3235.953	6869.239		
				5566.173	97.905	-3553.536		
				3970.065	-149.063	3323.592		
66	25	30	31	856.135	-5727.976	12306.576	-.096	3
				-22879.121	9394.656	3422.514		
				43753.600	1237.136	-6824.585		
				7243.336	1534.605	3161.536		

MARGIN OF SAFETY IN EACH LEG = .07989 -.84475 -.81063

ULTIMATE LOADS

NX	NY	NXY	MX	MY	MXY	TORQUE	GX	GY
769	0	77	0	0	0	0	0	0

OPTIMIZATION STEPS

```
IK =    0     SIDESTEPS =    0     STEPS =    2     CYCLES =    0
WT =       .00712     MARGIN =      -.84475     H(I) =        .75000      .50000      .03000      .03000      .03000
0.00000     0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =       .37985      -.84475      -.81063
     .04267       .04267       .04267

IK =    0     SIDESTEPS =    0     STEPS =    3     CYCLES =    0
WT =       .01361     MARGIN =      -.50115     H(I) =        .75000      .71119      .04267      .04267      .04267
0.00000     0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =       .83775      -.50115      -.40349
     .05336       .05336       .05336

IK =    0     SIDESTEPS =    0     STEPS =    4     CYCLES =    0
WT =       .02127     MARGIN =       .04963     H(I) =        .75000      .88939      .05336      .05336      .05336
0.00000     0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      1.56918       .07099       .04963
NEW OPTIMUM
 .75000
 .88939
 .05336
 .05336
 .05336
0.00000
0.00000
0.00000
0.00000
0.00000
 .50000
CALCULATION OF DERIVITIVES
     .05336       .05336       .05336

IK =    2     SIDESTEPS =    0     STEPS =    5     CYCLES =    1
WT =       .02124     MARGIN =       .04971     H(I) =        .75000      .88939      .05336      .05336      .05336
0.00000     0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      1.56849       .07130       .04971
    -.68603       .31503       .07983       .02257
     .05436       .05336       .05336

IK =    3     SIDESTEPS =    0     STEPS =    5     CYCLES =    1
WT =       .02141     MARGIN =       .05025     H(I) =        .75000      .88939      .05436      .05336      .05336
0.00000     0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      1.60826       .07215       .05025
   39.07828      1.15784       .01250       .14487
     .05336       .05436       .05436

IK =    4     SIDESTEPS =    0     STEPS =    7     CYCLES =    1
WT =       .02152     MARGIN =       .06608     H(I) =        .75000      .88939      .05336      .05436      .05436
0.00000     0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      1.57819       .13111       .06608
    9.00997     60.11690     19.04457       .25506
```

```
        .05336          .05436          .05436

    IK =    5     SIDESTEPS =    0     STEPS =    8     CYCLES =    1
    WT =       .02152      MARGIN =       .06868      M(I) =       .75000      .08835      .15536      .05436      .05436
    0.00000      0.00000      0.00000      0.00000      0.00000      .50000
    MARGIN(I) =      1.57815      .13111      .06866
       9.00997      60.11690      15.04457      .25506

SIDE STEPPING
    DIRECTION COSINES OF CONST. WT. SURFACE
    , CONSTRAINT SURFACES
    , MARGINS OF SAFETY
    , DIR. COSINES OF COMPOSITE CONSTRAINT SURFACE
    0.00000      .05625      .37384      .65454      .65454      0.00000      0.00000      0.00000      0.00000      0.00000
    0.00000
    0.00000      0.00000      0.00000
     .00060      .03371      .00296
     .95074      .01362      .02274
     .21920      .70704      .70692
     .21920      .70704      .70692
    0.00000      0.00000      0.00000
    0.00000      0.00000      0.00000
    0.00000      0.00000      0.00000
    0.00000      0.00000      0.00000
    0.00000      0.00000      0.00000
    0.00000      0.00000      0.00000
     .90000      .13111      .06866
    0.00000      .00316      .02036      .70696      .70696      0.00000      0.00000      0.00000      0.00000      0.00000
    0.00000
       .04790          .05838          .05833

    IK =   12     SIDESTEPS =    1     STEPS =    9     CYCLES =    1
    WT =       .02142      MARGIN =       .14012      M(I) =       .75000      .07504      .04795      .05536      .05536
    0.00000      0.00000      0.00000      0.00000      0.00000      .50000
    MARGIN(I) =      1.39858      .22162      .14012
       .04261          .06278          .06278

    IK =   12     SIDESTEPS =    2     STEPS =   10     CYCLES =    1
    WT =       .02140      MARGIN =       .21016      M(I) =       .75000      .05561      .04261      .06276      .06276
    0.00000      0.00000      0.00000      0.00000      0.00000      .50000
    MARGIN(I) =      1.17353      .29065      .21016
       .03776          .06699          .06699

    IK =   12     SIDESTEPS =    3     STEPS =   11     CYCLES =    1
    WT =       .02133      MARGIN =       .29241      M(I) =       .75000      .04435      .03776      .06699      .06699
    0.00000      0.00000      0.00000      0.00000      0.00000      .50000
    MARGIN(I) =      .97251      .36702      .29241
       .03335          .07101          .07101

    IK =   12     SIDESTEPS =    4     STEPS =   12     CYCLES =    1
    WT =       .02134      MARGIN =       .36315      M(I) =       .75000      .02936      .03335      .07101      .07101
    0.00000      0.00000      0.00000      0.00000      0.00000      .50000
    MARGIN(I) =      .77840      .43316      .36315
       .02937          .07465          .07465

    IK =   12     SIDESTEPS =    5     STEPS =   13     CYCLES =    1
```

```
WT =       .02137     MARGIN =       .42572     H(I) =       .75000      .81331      .02937      .07485      .07485
0.00000    0.00000    0.00000       0.00000    0.00000         .50000
MARGIN(I) =      .39743       .49545      .42572
     .02560       .07651       .07351

IK =  12     SIDESTEPS =   6      STEPS =  14      CYCLES =   1
WT =       .02135     MARGIN =       .42979     H(I) =       .75000      .79778      .02560      .07451      .07451
0.00000    0.00000    0.00000       0.00000    0.00000         .50000
MARGIN(I) =      .42979       .55437      .47111
     .02262       .08230       .05200

IK =  12     SIDESTEPS =   7      STEPS =  15      CYCLES =   1
WT =       .02135     MARGIN =       .03562     H(I) =       .75000      .76235      .02262      .08206      .08206
0.00000    0.00000    0.00000       0.00000    0.00000         .50000
MARGIN(I) =      .03562       .61042      .51351
DESCENDING
     .02560       .07651       .07651

IK =  12     SIDESTEPS =   8      STEPS =  16      CYCLES =   1
WT =       .02135     MARGIN =       .42979     H(I) =       .75000      .79778      .02560      .07651      .07651
0.00000    0.00000    0.00000       0.00000    0.00000         .50000
MARGIN(I) =      .42979       .55437      .47111
     .02165       .05642       .05642

IK =  12     SIDESTEPS =   8      STEPS =  17      CYCLES =   1
WT =       .01533     MARGIN =      -.00143     H(I) =       .75000      .77761      .02165      .05642      .05642
0.00000    0.00000    0.00000       0.00000    0.00000         .50000
MARGIN(I) =     -.00143       .12157      .07223
     .02231       .05943       .05943

IK =  12     SIDESTEPS =   8      STEPS =  18      CYCLES =   1
WT =       .01622     MARGIN =      -.03362     H(I) =       .75000      .78150      .02231      .05943      .05943
0.00000    0.00000    0.00000       0.00000    0.00000         .50000
MARGIN(I) =     -.03362       .15061      .12700
NEW OPTIMUM
 .75000
 .78150
 .02231
 .05943
 .05943
0.00000
0.00000
0.00000
0.00000
0.00000
 .50000
CALCULATION OF DERIVITIVES
     .02231       .05943       .05943

IK =   2     SIDESTEPS =   0      STEPS =  19      CYCLES =   2
WT =       .01624     MARGIN =      -.00496     H(I) =       .75000      .78250      .02231      .05943      .05943
0.00000    0.00000    0.00000       0.00000    0.00000         .50000
MARGIN(I) =     -.00496       .15094      .12700
  -1.33248       .13318       .00117       .01927
     .02331       .05943       .05943

IK =   3     SIDESTEPS =   0      STEPS =  20      CYCLES =   2
```

```
WT =        .01655      MARGIN =      -.43235     H(I) =       .75000       .76150       .02331       .05543       .05543
0.00000      0.00000      0.00000      0.00000      0.00000      .50000
MARGIN(I) =       -.46235        .18288        .13173
-478.72999          2.07445          4.00405          .12635
     .02231          .06043          .06043

IK =    4      SIDESTEPS =    0      STEPS =   21      CYCLES =    2
WT =        .01644      MARGIN =      -.00628     H(I) =       .75000       .76150       .02231       .06043       .06043
0.00000      0.00000      0.00000      0.00000      0.00000      .50000
MARGIN(I) =       -.00628        .19988        .14425
  -2.65617        19.07406        17.16771          .22472
    .02231          .06043          .06043

IK =    5      SIDESTEPS =    0      STEPS =   22      CYCLES =    2
WT =        .01644      MARGIN =      -.00628     H(I) =       .75000       .76150       .02231       .06043       .06043
0.00000      0.00000      0.00000      0.00000      0.00000      .50000
MARGIN(I) =       -.00628        .19988        .14425
  -2.65617        19.07406        17.16771          .22472

SIDE STEPPING
    DIRECTION COSINES OF CONST. WT. SURFACE
    , CONSTRAINT SURFACES
    , MARGINS OF SAFETY
    , DIR. COSINES OF COMPOSITE CONSTRAINT SURFACE
0.00000      .05615      .37332      .65462      .65462      0.00000      0.00000      0.00000      0.00000      0.00000
0.00000
0.00000      0.00000      0.00000
 .50000      .03492      .00605
 .50000      .07668      .18844
 .50000      .70562      .69444
 .50000      .70562      .69444
0.00000      0.00000      0.00000
0.00000      0.00000      0.00000
0.00000      0.00000      0.00000
0.00000      0.00000      0.00000
0.00000      0.00000      0.00000
0.00000      0.00000      0.00000
-.00628      .19988      .14425
0.00000      .45036      .47361      .53264      .53264      0.00000      0.00000      0.00000      0.00000      0.00000
0.00000
    .02210          .05404          .05404

IK =   12      SIDESTEPS =    1      STEPS =   23      CYCLES =    2
WT =        .01630      MARGIN =      -.00165     H(I) =       .75000       .65544       .02210       .05404       .05404
0.00000      0.00000      0.00000      0.00000      0.00000      .50000
MARGIN(I) =       -.00165        .08731        .0534
    .02230          .05829          .05323

IK =   12      SIDESTEPS =    2      STEPS =   24      CYCLES =    2
WT =        .01632      MARGIN =      -.0244      H(I) =       .75000       .60000       [illegible]       .05825       .05825
0.00000      0.00000      0.00000      0.00000      0.00000      .50000
MARGIN(I) =       -.32443        .16153        .1.44
DESCENDING
    .02231          .05543          .05543

IK =   12      SIDESTEPS =    3      STEPS =   25      CYCLES =    2
```

```
WT =        .01622      MARGIN =       -.11562      M(I) =        .75441     .76150     .02231     .05943     .05943
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =       -.00362      .18081      .12706
CALCULATION OF DERIVITIVES
   .02231          .05943          .05943

IK =    2      SIDESTEPS =    0      STEPS =  29      CYCLES =    3
WT =        .01624      MARGIN =       -.03496      M(I) =        .75000     .73250     .02231     .05943     .05943
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =       -.00496      .10394      .12706
  -1.33248        .13318          .03117          .01927
   .02331         .05943          .05943

IK =    3      SIDESTEPS =    0      STEPS =  27      CYCLES =    3
WT =        .01635      MARGIN =       -.46235      M(I) =        .75000     .78150     .02331     .05943     .05943
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =       -.46235      .10268      .13173
 -478.72999       2.07445         4.60405         .12835
   .02231         .06043          .06043

IK =    4      SIDESTEPS =    0      STEPS =  28      CYCLES =    3
WT =        .01644      MARGIN =       -.00628      M(I) =        .75000     .75150     .02231     .06043     .06043
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =       -.00628      .13988      .14429
  -2.65617       19.07406        17.18771         .22472
   .02231         .06043          .06043

IK =    5      SIDESTEPS =    0      STEPS =  23      CYCLES =    3
WT =        .01644      MARGIN =       -.00628      M(I) =        .75000     .75150     .02231     .06043     .06043
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =       -.00628      .13988      .14429
  -2.65617       19.07406        17.18771         .22472

SIDE STEPPING
  DIRECTION COSINES OF CONST. WT. SURFACE
    , CONSTRAINT SURFACES
    , MARGINS OF SAFETY
    , DIR. COSINES OF COMPOSITE CONSTRAINT SURFACE
0.00000      .05615       .37386       .65462       .65462     0.00000     0.00000     0.00000     0.00000     0.00000
0.00000
0.00000     0.00000     0.00000
 .50000      .00492       .00005
 .50000      .37683       .18844
 .50000      .70502       .69444
 .50000      .70502       .69444
0.00000     0.00000     0.00000
0.00000     0.00000     0.00000
0.00000     0.00000     0.00000
0.00000     0.00000     0.00000
0.00000     0.00000     0.00000
0.00000     0.00000     0.00000
-.00628      .13988       .14425
0.00000      .45638       .47361       .53264       .53264     0.00000     0.00000     0.00000     0.00000     0.00000
0.00000
  .02239          .05924          .05924
```

```
IK =  12     SIDESTEPS =   1     STEPS =  30     CYCLES =   3
WT =        .01633      MARGIN =       -.00409      H(1) =        .75000     .79132     .02239     .05924     .05924
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =       -.00409       .17069       .12461
    .02240            .05968          .05965

IK =  12     SIDESTEPS =   2     STEPS =  31     CYCLES =   3
WT =        .01634      MARGIN =        .00357      H(1) =        .75000     .78423     .02240     .05968     .05968
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =        .00357        .13003       .13171
    .02239            .05934          .05934

IK =  12     SIDESTEPS =   3     STEPS =  32     CYCLES =   3
WT =        .01633      MARGIN =       -.00412      H(1) =        .75000     .78643     .02239     .05934     .05934
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =       -.00412       .17985       .12576
DESCENDING
    .02240            .05968          .05965

IK =  12     SIDESTEPS =   4     STEPS =  33     CYCLES =   3
WT =        .01634      MARGIN =        .00357      H(1) =        .75000     .78429     .02240     .05968     .05968
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =        .00357        .16003       .13171
CALCULATION OF DERIVITIVES
    .02241            .05968          .05965

IK =   2     SIDESTEPS =   0     STEPS =  34     CYCLES =   4
WT =        .01630      MARGIN =        .00230      H(1) =        .75000     .78529     .02240     .05968     .05968
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =        .00230        .13021       .13171
  -1.31676           .13419          .31253           .01936
    .02340            .35968          .35963

IK =   3     SIDESTEPS =   0     STEPS =  35     CYCLES =   4
WT =        .01647      MARGIN =        .13637      H(1) =        .75000     .78429     .02340     .05968     .05968
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =        .15324        .13616       .13637
 149.56973          2.08895         4.06034           .12877
    .02240            .06008          .06065

IK =   4     SIDESTEPS =   0     STEPS =  36     CYCLES =   4
WT =        .01650      MARGIN =        .00035      H(1) =        .75000     .78429     .02240     .06068     .06068
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =        .00355        .20515       .14890
  -2.71975          19.07450         17.13743          .22546
    .02240            .06008          .06065

IK =   5     SIDESTEPS =   0     STEPS =  37     CYCLES =   4
WT =        .01650      MARGIN =        .00035      H(1) =        .75000     .78429     .02240     .06068     .06068
0.00000     0.00000     0.00000     0.00000     0.00000     .50000
MARGIN(I) =        .00335        .20515       .14890
  -2.71976          19.07460         17.13749          .22546

SIDE STEPPING
    DIRECTION COSINES OF CONST. WT. SURFACE
    , CONSTRAINT SURFACES
```

```
, MARGINS OF SAFETY
, DIR. COSINES OF COMPOSITE CONSTRAINT SURFACE
0.00000      .05620      .37358      .65462      .65462      0.00000      0.00000      0.00000      0.00000      0.00000
0.00000
0.00000      0.00000     0.00000
.30000       .03498      .00000
1.00000      .07721      .15845
.30000       .70499      .69445
.00000       .70499      .69445
0.00000      0.00000     0.00000
0.00000      0.00000     0.00000
0.00000      0.00000     0.00000
0.00000      0.00000     0.00000
0.00000      0.00000     0.00000
0.00000      0.00000     0.00000
.00055       .20015      .14850
0.00000      .00017      .99300      .33336      .08336      0.00000      0.00000      0.00000      0.00000      0.00000
0.00000
    .02245        .05970        .05970

IK =  12     SIDESTEPS =   1     STEPS =  38     CYCLES =   4
WT =      .01635     MARGIN =      .01601    M(I) =      .75000     .70429     .02248     .05970     .05970
0.00000      0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      .01601      .13063      .13245
    .02257        .05972        .05972

IK =  12     SIDESTEPS =   2     STEPS =  39     CYCLES =   4
WT =      .01637     MARGIN =      .02355    M(I) =      .75000     .70429     .02257     .05972     .05972
0.00000      0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      .02655      .10719      .13326
    .02266        .05974        .05974

IK =  12     SIDESTEPS =   3     STEPS =  40     CYCLES =   4
WT =      .01639     MARGIN =      .04115    M(I) =      .75000     .70429     .02266     .05974     .05974
0.00000      0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      .04115      .13774      .13594
    .02275        .05976        .05976

IK =  12     SIDESTEPS =   4     STEPS =  41     CYCLES =   4
WT =      .01640     MARGIN =      .05596    M(I) =      .75000     .70429     .02275     .05976     .05976
0.00000      0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      .05596      .13636      .13469
    .02283        .05977        .05977

IK =  12     SIDESTEPS =   5     STEPS =  42     CYCLES =   4
WT =      .01642     MARGIN =      .06694    M(I) =      .75000     .70429     .02283     .05977     .05977
0.00000      0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      .06694      .13665      .13544
    .02292        .05979        .05979

IK =  12     SIDESTEPS =   6     STEPS =  43     CYCLES =   4
WT =      .01643     MARGIN =      .08039    M(I) =      .75000     .70429     .02292     .05979     .05979
0.00000      0.00000     0.00000     0.00000     0.00000      .50000
MARGIN(I) =      .08039      .13941      .13619
    .02301        .05981        .05981

IK =  12     SIDESTEPS =   7     STEPS =  44     CYCLES =   4
```

```
WT =        .01645    MARGIN =      .09341      H(I) =       .75000      .76423      .02301      .05981      .05981
0.00000     0.00000     0.00000     0.00000     0.00000       .50000
MARGIN(I) =        .09341      .16997      .13693
     .02310          .05983          .05983

IK =  12      SIDESTEPS =   8      STEPS =  45      CYCLES =   4
WT =        .01645    MARGIN =      .13030      H(I) =       .75000      .76429      .02310      .05983      .05983
0.00000     0.00000     0.00000     0.00000     0.00000       .50000
MARGIN(I) =        .12030      .19053      .13709
     .02319          .05985          .05985

IK =  12      SIDESTEPS =   9      STEPS =  46      CYCLES =   4
WT =        .01646    MARGIN =      .12050      H(I) =       .75000      .73429      .02319      .05985      .05985
0.00000     0.00000     0.00000     0.00000     0.00000       .50000
MARGIN(I) =        .12036      .19108      .13044
     .02328          .05987          .05987

IK =  12      SIDESTEPS =  10      STEPS =  47      CYCLES =   4
WT =        .01650    MARGIN =      .13440      H(I) =       .75000      .76429      .02328      .05987      .05987
0.00000     0.00000     0.00000     0.00000     0.00000       .50000
MARGIN(I) =        .13440      .13164      .13319
DESCENDING
     .02313          .05985          .05985

IK =  12      SIDESTEPS =  11      STEPS =  43      CYCLES =   4
WT =        .01648    MARGIN =      .12056      H(I) =       .75000      .76429      .02319      .05985      .05985
0.00000     0.00000     0.00000     0.00000     0.00000       .50000
MARGIN(I) =        .12036      .19108      .13944
     .02214          .05513          .05513

IK =  12      SIDESTEPS =  11      STEPS =  49      CYCLES =   4
WT =        .01519    MARGIN =     -.01075      H(I) =       .75000      .77898      .02214      .05513      .05513
0.00000     0.00000     0.00000     0.00000     0.00000       .50000
MARGIN(I) =       -.01075      .09800      .05220
NEW OPTIMUM
 .75000
 .77898
 .02214
 .05513
 .05513
0.00000
0.00000
0.00000
0.00000
0.00000
 .50000
```

OPTIMIZED LAYER THICKNESSES

1	.75000
2	.77898
3	.02214
4	.05513
5	.05513
6	0.00000
7	0.00000
8	0.00000
9	0.00000
10	0.00000
11	.50000

FLAT PANEL WIDTH = 2.16626

LENGTH = 2.62500

FIRST VERTICAL PLATE = 1 SKEW A = 25 SKEW B = 45 LWR FACE = 0 UPPER FACE = 0

```
NUMBER OF NODAL POINTS------  32
NUMBER OF ELEMENTS----------  60
NUMBER OF LOAD CASES      ---  1
```

NODAL POINT	X	Y	Z
1	0.0000	.0433	0.0000
2	0.0000	1.9493	0.0000
3	0.0000	.2165	.7790
4	0.0000	1.5164	.7790
5	.3750	0.0000	0.0000
6	.3750	1.2993	0.0000
7	.3750	.3664	.7790
8	.3750	2.1663	.7790
9	.7500	.0433	0.0000
10	.7500	1.9498	0.0000
11	.7500	.2165	.7790
12	.7500	1.5164	.7790
13	1.1250	0.0000	0.0000
14	1.1250	1.2993	0.0000
15	1.1250	.3664	.7790
16	1.1250	2.1663	.7790
17	1.5000	.0433	0.0000
18	1.5000	1.9498	0.0000
19	1.5000	.2165	.7790
20	1.5000	1.5164	.7790
21	1.8750	0.0000	0.0000
22	1.8750	1.2993	0.0000
23	1.3750	.3664	.7790
24	1.8750	2.1663	.7790
25	2.2500	.0433	0.0000
26	2.2500	1.9498	0.0000
27	2.2500	.2165	.7790
28	2.2500	1.5164	.7790
29	2.6250	0.0000	0.0000
30	2.6250	1.2993	0.0000
31	2.6250	.3664	.7790
32	2.6250	2.1663	.7790

LOAD CASE 1

NEW NODE	OLD NODE	UX	UY	UZ
1	1	-60.1746	-6.0175	0.0000
2	2	-52.6829	-6.0175	0.0000
3	3	-67.4662	-6.0175	0.0000
4	3001	-269.3656	-24.6693	0.0000
5	4	-60.1746	-6.0175	0.0000
6	4002	-240.6983	-24.6693	0.0000
7	5	-67.4662	-6.0175	0.0000
8	5003	-269.3656	-24.6692	0.0000
9	6	-60.1746	-6.0175	0.0000
10	6004	-240.6983	-24.6693	0.0000
11	7	-60.1746	-6.0175	0.0000
12	7001	-240.6983	-24.6693	0.0000
13	7006	-240.6983	-24.6693	0.0000
14	8	-52.6829	-6.0175	0.0000
15	8002	-211.5317	-24.6693	0.0000
16	9	0.0000	0.0000	0.0000
17	9007	0.0000	0.0000	0.0000
18	10	7.2917	0.0000	0.0000
19	10008	0.0000	0.0000	0.0000
20	11	-7.2917	0.0000	0.0000
21	11005	-269.3656	0.0000	0.0000
22	11009	0.0000	0.0000	0.0000
23	12	0.0000	0.0000	0.0000
24	12006	0.0000	0.0000	0.0000
25	12010	0.0000	0.0000	0.0000
26	13	-7.2917	0.0000	0.0000
27	13011	-29.1667	0.0000	0.0000
28	14	0.0000	0.0000	0.0000
29	14012	0.0000	0.0000	0.0000
30	15	0.0000	0.0000	0.0000
31	15009	0.0000	0.0000	0.0000
32	15014	0.0000	0.0000	0.0000
33	16	7.2917	0.0000	0.0000
34	16010	29.1667	0.0000	0.0000
35	17	0.0000	0.0000	0.0000
36	17015	0.0000	0.0000	0.0000
37	18	7.2917	0.0000	0.0000
38	18016	29.1667	0.0000	0.0000
39	19	-7.2917	0.0000	0.0000
40	19013	-29.1667	0.0000	0.0000
41	19017	0.0000	0.0000	0.0000
42	20	0.0000	0.0000	0.0000
43	20014	0.0000	0.0000	0.0000
44	20018	0.0000	0.0000	0.0000
45	21	-7.2917	0.0000	0.0000
46	21019	-29.1667	0.0000	0.0000
47	22	0.0000	0.0000	0.0000
48	22020	0.0000	0.0000	0.0000
49	23	0.0000	0.0000	0.0000
50	23017	0.0000	0.0000	0.0000
51	23022	0.0000	0.0000	0.0000
52	24	7.2917	0.0000	0.0000

53	24018	29.1667	0.0000	0.0000	
54	25	60.1746	6.0175	0.0000	
55	25023	0.0000	0.0000	0.0000	
56	26	67.4662	6.0175	0.0000	
57	26024	29.1667	0.0000	0.0000	
58	27	52.8829	6.0175	0.0000	
59	27021	0.0000	0.0000	0.0000	
60	27025	211.5317	24.0698	0.0000	
61	28	60.1746	6.0175	0.0000	
62	28022	0.0000	0.0000	0.0000	
63	28026	240.6983	24.0698	0.0000	
64	29	52.8829	6.0175	0.0000	
65	29027	211.5317	24.0698	0.0000	
66	30	60.1746	6.0175	0.0000	
67	30028	240.6983	24.0698	0.0000	
68	31	60.1746	6.0175	0.0000	
69	31025	240.6983	24.0698	0.0000	
70	31030	240.6983	24.0698	0.0000	
71	32	67.4662	6.0175	0.0000	
72	32026	269.8650	24.0693	0.0000	

ELEMENT NUMBER	I	J	K	EX	EY	EXY	G	T	LF	LG	LR	LS	LT
1	3	11	5	4031496	503937	125984	500000	.0221	3	20	7	21	8
2	5	13	11	4031496	503937	125984	500000	.0221	7	26	22	27	21
3	11	19	13	4031496	503937	125984	500000	.0221	20	39	26	40	27
4	13	21	19	4031496	503937	125984	500000	.0221	26	45	39	46	40
5	19	27	21	4031496	503937	125984	500000	.0221	39	58	45	59	46
6	21	29	27	4031496	503937	125984	500000	.0221	45	64	58	65	59
7	1	9	7	4031496	503937	125984	500000	.0221	1	16	11	17	12
8	7	15	9	4031496	503937	125984	500000	.0221	11	30	16	31	17
9	9	17	15	4031496	503937	125984	500000	.0221	16	35	30	36	31
10	15	23	17	4031496	503937	125984	500000	.0221	30	49	35	50	36
11	17	25	23	4031496	503937	125984	500000	.0221	35	54	49	55	50
12	23	31	25	4031496	503937	125984	500000	.0221	49	68	54	69	55
13	4	12	6	4031496	503937	125984	500000	.0221	5	23	9	24	10
14	6	14	12	4031496	503937	125984	500000	.0221	9	28	23	29	24
15	12	20	14	4031496	503937	125984	500000	.0221	23	42	28	43	29
16	14	22	20	4031496	503937	125984	500000	.0221	28	47	42	48	43
17	20	28	22	4031496	503937	125984	500000	.0221	42	61	47	62	48
18	22	30	28	4031496	503937	125984	500000	.0221	47	66	61	67	62
19	2	10	8	4031496	503937	125984	500000	.0221	2	18	14	19	15
20	8	16	10	4031496	503937	125984	500000	.0221	14	33	18	34	19
21	10	18	16	4031496	503937	125984	500000	.0221	18	37	33	38	34
22	16	24	18	4031496	503937	125984	500000	.0221	33	52	37	53	38
23	18	26	24	4031496	503937	125984	500000	.0221	37	56	52	57	53
24	24	32	26	4031496	503937	125984	500000	.0221	52	71	56	72	57
25	1	5	3	4031496	503937	125984	500000	.0551	1	7	3	8	4
26	2	6	4	4031496	503937	125984	500000	.0551	2	9	5	10	6
27	4	7	6	4031496	503937	125984	500000	.0551	5	14	9	15	10
28	6	9	7	4031496	503937	125984	500000	.0551	9	16	11	17	13
29	7	11	9	4031496	503937	125984	500000	.0551	11	20	16	22	17
30	9	13	11	4031496	503937	125984	500000	.0551	16	26	22	27	22
31	8	12	10	4031496	503937	125984	500000	.0551	14	23	18	25	19
32	10	14	12	4031496	503937	125984	500000	.0551	18	26	23	29	25
33	12	15	14	4031496	503937	125984	500000	.0551	23	30	26	32	29
34	14	17	15	4031496	503937	125984	500000	.0551	28	35	30	36	32
35	15	19	17	4031496	503937	125984	500000	.0551	30	39	35	41	36
36	17	21	19	4031496	503937	125984	500000	.0551	35	45	39	46	41
37	16	20	18	4031496	503937	125984	500000	.0551	33	42	37	44	38
38	18	22	20	4031496	503937	125984	500000	.0551	37	47	42	43	44
39	20	23	22	4031496	503937	125984	500000	.0551	42	49	47	51	48
40	22	25	23	4031496	503937	125984	500000	.0551	47	54	49	55	51
41	23	27	25	4031496	503937	125984	500000	.0551	49	58	54	60	55
42	25	29	27	4031496	503937	125984	500000	.0551	54	64	58	65	60
43	24	28	26	4031496	503937	125984	500000	.0551	52	61	56	63	57
44	26	30	28	4031496	503937	125984	500000	.0551	56	66	61	67	63
45	28	31	30	4031496	503937	125984	500000	.0551	61	68	66	70	67
46	4	8	2	4031496	503937	125984	500000	.0551	5	14	2	15	6
47	3	7	1	4031496	503937	125984	500000	.0551	3	11	1	12	4
48	1	6	7	4031496	503937	125984	500000	.0551	1	9	11	13	12
49	7	12	6	4031496	503937	125984	500000	.0551	11	23	9	24	13
50	6	10	12	4031496	503937	125984	500000	.0551	9	18	23	25	24

ELEMENT NUMBER	I	J	K	EX	EY	EXY	G	T	LF	LG	LR	LS	LT
51	12	10	10	4031496	503937	125984	500000	.0351	23	33	16	34	25
52	5	9	11	4031496	503937	125984	500000	.0351	7	16	20	22	21
53	11	15	9	4031496	503937	125984	500000	.0351	20	30	16	31	22
54	9	14	15	4031496	503937	125984	500000	.0351	16	26	30	32	31
55	15	20	14	4031496	503937	125984	500000	.0351	30	42	26	43	32
56	14	16	20	4031496	503937	125984	500000	.0351	26	37	42	44	43
57	20	24	16	4031496	503937	125984	500000	.0351	42	52	37	53	44
58	13	17	19	4031496	503937	125984	500000	.0351	26	35	39	41	40
59	19	25	17	4031496	503937	125984	500000	.0351	39	49	35	50	41
60	17	22	23	4031496	503937	125984	500000	.0351	35	47	45	51	50
61	23	28	22	4031496	503937	125984	500000	.0351	49	61	47	62	51
62	22	26	25	4031496	503937	125984	500000	.0351	47	56	61	63	62
63	28	32	26	4031496	503937	125984	500000	.0351	61	71	56	72	63
64	21	25	27	4031496	503937	125984	500000	.0351	45	54	56	60	55
65	27	31	25	4031496	503937	125984	500000	.0351	53	66	54	69	60
66	25	30	31	4031496	503937	125984	500000	.0351	54	66	66	73	69

LEO BUCKLING STRESS

DEFLECTIONS LOAD CASE 1

NODAL POINT	UX	UY	UZ
1	-1.0369106E-02	6.2112945E-03	-1.3853273E-02
2	-2.0369603E-03	-2.9772085E-04	-1.2248492E-02
3	-2.2873086E-02	3.8871955E-03	-7.7159[illegible]E-03
3001	-2.7047883E-02	6.6100015E-03	-4.9936573E-03
4	-1.5045460E-02	-4.4396514E-03	-3.4417011E-03
4002	-1.8807640E-02	-4.7010239E-03	-1.1749752E-02
5	-1.3378536E-02	6.5812406E-03	-1.2927[illegible]E-03
5003	-1.8099928E-02	1.1264314E-02	-5.4545[illegible]E-03
6	-3.2175883E-03	-1.3600558E-04	-6.3735[illegible]E-03
6004	-7.2439215E-03	1.4271734E-03	-7.6087[illegible]E-03
7	-1.4286604E-02	-1.3244731E-03	-2.4099547E-03
7001	-8.9163586E-03	-3.0747146E-03	-1.1349594E-02
7006	-1.1615445E-02	-1.1362441E-03	-7.4342[illegible]E-03
8	-1.0270876E-02	-7.4435250E-03	-1.2779596E-02
8002	-2.2868167E-03	-1.3619332E-02	-1.1579[illegible]E-02
9	-2.7161992E-15	3.2568091E-16	-1.5521[illegible]E-15
9007	-6.0601551E-03	1.2029645E-03	-4.7451[illegible]E-03
10	3.0416652E-03	-3.7492084E-03	-4.0089[illegible]E-03
10008	-3.2561430E-04	-3.2415[illegible]E-03	-7.3576[illegible]E-03
11	-1.4306673E-02	6.6705049E-04	-1.8181[illegible]E-02
11005	-1.5931008E-02	5.5291260E-03	-1.7356[illegible]E-03
11009	-7.3552816E-03	6.2753336E-04	-1.4076[illegible]E-02
12	-7.3364052E-03	-5.0890300E-03	-4.5305[illegible]E-02
12006	-4.8070768E-03	-4.1805352E-03	-5.7441[illegible]E-03
12010	-3.2034403E-03	-4.0390559E-03	-4.3552[illegible]E-03
13	-5.5042972E-03	7.5306579E-04	1.2132[illegible]E-02
13011	-1.2185136E-02	-2.1493387E-03	1.7170[illegible]E-04
14	2.0406842E-03	-2.0102993E-03	4.1367[illegible]E-07
14012	-1.8601193E-03	-3.3552[illegible]E-03	-2.0483[illegible]E-02
15	-6.0338405E-03	-3.2047701E-03	-3.7757[illegible]E-04
15009	-3.5135069E-03	-1.2549675E-03	-5.5702[illegible]E-04
15014	-1.4247689E-03	-2.5443522E-03	-5.5944[illegible]E-05
16	-1.7711365E-03	-3.1435755E-03	-6.5319[illegible]E-04
16010	2.2501542E-03	-7.6770363E-03	-2.3491[illegible]E-03
17	2.5018724E-03	-5.2508341E-16	1.0521[illegible]E-15
17015	-1.1026691E-03	-1.7490550E-03	6.5016[illegible]E-04
18	9.0545675E-03	-4.7105521E-03	3.1127[illegible]E-03
18016	4.9586333E-03	-5.6646849E-03	1.3496[illegible]E-02
19	-7.7007509E-03	-2.9532340E-03	3.9959[illegible]E-04
19013	-8.7430675E-03	2.4100916E-03	4.0395[illegible]E-04
19017	-1.8930629E-03	-1.5297179E-03	1.1016422E-03
20	-3.0032125E-04	-6.7261821E-03	2.3597[illegible]E-03
20014	1.1444987E-03	-4.0718071E-03	1.1820[illegible]E-02
20018	3.8238431E-03	-5.6412095E-03	2.7244915E-03
21	7.8615951E-04	1.2130121E-03	4.4138[illegible]E-03
21019	-5.5302523E-03	-3.3438650E-03	2.3351[illegible]E-03
22	6.3859774E-03	-3.5676529E-03	5.9314[illegible]E-03
22020	4.1684672E-03	-5.2856710E-03	4.1419[illegible]E-03
23	-1.4974603E-04	-6.2949885E-03	4.4154[illegible]E-03
23017	1.3061681E-03	-3.2614444E-03	2.3257[illegible]E-03

DEFLECTIONS LOAD CASE 1

NODAL POINT	UX	UY	UZ
23022	4.58781452E-03	-4.30575360E-03	5.34555580E-03
24	6.15379835E-03	-1.17162345E-02	5.32633378E-03
24018	7.72841795E-03	-7.93433306E-03	4.65770036E-03
25	9.98154995E-03	-2.30422026E-03	6.27255326E-03
25023	3.35692018E-03	-5.24465366E-03	5.97305676E-03
26	1.57118652E-02	-7.61403306E-03	1.07275066E-02
26024	1.33739926E-02	-1.22203746E-02	6.24175166E-03
27	-1.59646452E-03	-5.76641306E-03	5.20142356E-03
27021	-2.74986146E-03	2.32451355E-03	4.37345512E-03
27025	7.50598406E-03	-3.24487695E-03	6.96815136E-03
28	7.74176535E-03	-1.11119295E-02	7.98302376E-03
28022	7.35722226E-03	-5.96073786E-03	7.36355406E-03
28026	1.49291506E-02	-9.37332616E-03	9.30525356E-03
29	2.78479886E-03	-6.34355869E-03	8.17005146E-03
29027	-6.11749652E-04	-7.71600206E-03	6.74305516E-03
30	1.60269006E-02	-6.87242262E-03	9.81757495E-03
30028	9.45770809E-03	-1.30197536E-02	8.61791566E-03
31	6.70619436E-03	-1.64571306E-02	1.32321746E-02
31025	5.99946136E-03	-2.12812056E-03	1.16524366E-02
31030	1.82360936E-02	-3.53656006E-03	1.15233376E-02
32	9.44240595E-03	-1.14374335E-02	1.56165566E-02
32026	1.25674736E-02	-5.52245976E-03	1.26211566E-02

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
1	3	11	5	43464.604	-1356.729	-4343.[illegible]	.344	1
				46611.317	-63.356	-703.37[illegible]		
				320.047	-1366.097	5533.76		
				36265.443	-1000.001	-4646.141		
2	5	13	11	32277.007	234.715	-7726.37	.263	1
				31647.323	-504.073	-7023.11		
				37620.924	-347.463	1234.536		
				46564.456	-255.656	-4563.59		
3	11	19	13	34153.465	6000.571	-3259.47	.363	1
				37002.152	-366.364	-2706.13		
				37756.172	-57.735	5357.025		
				36615.256	-143.463	-226.44		
4	13	21	19	24979.067	447.566	-7133.71	.434	1
				40762.350	-2373.206	-4515.77		
				35433.400	63.143	4460.36		
				33731.791	-6220.326	-3736.37		
7	1	9	7	65127.163	-1552.766	6203.436	-.611	1
				45643.236	1665.561	6551.[illegible]		
				-7667.061	-126.355	1526.75		
				34467.453	-1646.572	4370.455		
8	7	15	9	46176.765	-447.566	-656.507	1.346	1
				41576.432	276.625	2161.22		
				16150.567	-2776.961	2642.556		
				34627.350	-362.345	1395.322		
9	9	17	15	6491.067	-1755.651	-4513.52	1.175	1
				20645.055	1157.532	-3933.25		
				43262.623	-44.955	-1491.13		
				23240.243	-215.676	-3412.033		
10	15	23	17	33915.063	-542.163	2465.723	1.622	1
				26669.475	-172.647	363.27		
				21577.746	2762.656	606.216		
				26167.556	562.606	1323.159		
13	4	12	6	47623.755	-356.175	6105.331	.426	1
				29553.344	2142.366	46444.73		
				13456.356	-356.525	5311.36		
				33622.362	511.227	4657.113		
14	6	14	12	33667.592	-1732.065	-165.546	2.135	1
				31331.553	553.966	1411.590		
				30156.725	3.102	-57.325		
				31465.363	-376.551	355.73		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
15	12	20	14	41157.142	-269.658	617.145	1.643	1
				39215.557	160.902	775.794		
				24308.274	-97.840	3.533		
				34693.058	-[illegible]8.677	405.524		
16	14	22	20	31819.223	158.990	-568.725	2.191	1
				33849.052	-42.934	-663.757		
				31689.013	-110.047	-200.[illegible]80		
				32439.295	15.356	-566.[illegible]04		
19	2	10	8	37611.734	-5022.541	2666.[illegible]99	.171	1
				22330.921	3105.457	8815. 03		
				10045.313	-2830.074	-779.[illegible]41		
				23595.969	-1762.093	3574. 54		
20	8	16	10	45231.766	-3279.515	2101.552	.746	1
				41550.745	1775.132	4626.756		
				95[illegible].493	-409.187	-1280.459		
				33262.966	-656.973	1816. 93		
21	10	18	16	31786.314	-1364.115	2675.[illegible]96	1.519	1
				25986.592	1173.240	2329.149		
				26950.[illegible]59	151.934	-1257.325		
				26964.322	166.554	1250. 73		
25	1	5	3	-17210.293	6915.709	-5364.128	.096	2
				-4321.115	-4577.767	-3309.319		
				26726.172	719.178	8150.550		
				-268.412	1319.040	-2166.794		
26	2	6	4	-8110.727	3365.806	-1333. 17	.557	2
				661.417	-1366.486	-6697. 13		
				14679.213	2904.656	2390.666		
				2450.030	1621.309	-1306.[illegible]35		
27	4	7	6	-11570.373	2445.147	-1523.135	2.664	2
				-8627.518	-2068.155	-2345.129		
				10685.160	370.149	1744.126		
				-3664.532	249.337	-726.366		
28	6	9	7	4647.545	1315.167	-5837.093	1.752	2
				11553.514	-2366.633	-1666.156		
				1390.559	-924.513	55.[illegible]56		
				5677.349	-657.326	-[illegible]3[illegible]2.564		
29	7	11	9	-14252.065	-[illegible]69.2[illegible]	-310[illegible]. 56	2.144	2
				-9340.445	-56[illegible].[illegible]	-125[illegible].752		
				14151.517	-2152.965	130[illegible].[illegible]		
				-3149.097	-655.639	-1[illegible]		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
30	9	13	11	-18824.938	-2461.454	-1243.254	1.386	2
				-17569.363	2177.604	-623.225		
				18237.429	-327.714	1463.435		
				-6052.290	-103.838	-153.331		
31	8	12	10	-9437.190	-614.406	-4193.325	1.566	2
				297.671	-846.940	-1779.146		
				-558.044	-371.560	-1564.426		
				-3232.521	-610.509	-2567.630		
32	10	14	12	-10615.022	267.319	37.141	3.552	2
				-10294.180	-223.859	51.920		
				447.375	-57.462	91.951		
				-6626.611	2.626	76.374		
33	12	15	14	-4324.830	-49.621	-152.758	6.312	2
				-3486.469	-19.417	-81.436		
				-6127.388	26.978	-549.437		
				-4646.890	-14.667	-261.576		
34	14	17	15	-11544.051	66.933	414.059	2.799	2
				-8362.272	1791.212	-2393.132		
				-1649.030	246.694	-457.303		
				-7185.318	655.013	-812.309		
35	15	19	17	-4218.645	-539.504	2165.591	3.421	2
				-7139.123	-72.191	146.134		
				-5484.032	3029.167	-229.534		
				-5613.733	665.751	674.117		
40	4	8	2	-8954.136	7357.430	-3294.591	.852	3
				6808.353	-7233.272	-3626.449		
				8651.934	-574.748	3397.242		
				2168.715	14.462	-2841.356		
47	3	7	1	-6953.962	4336.774	-1127.458	.508	3
				682.143	-2513.645	-7308.731		
				18462.069	2695.117	3374.377		
				4663.017	1767.082	-1337.237		
48	1	6	7	-14422.481	2359.068	-3566.347	1.585	3
				-5953.460	-2253.106	-3711.436		
				17203.119	-2.031	1791.441		
				-1657.067	34.843	-1828.914		
49	7	12	6	-912.465	667.657	-3391.337	2.262	3
				4935.326	-958.997	-1651.322		
				14613.792	-1163.269	1565.071		
				6612.213	-456.133	-1226.203		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LCG
50	6	10	12	-3059.995	-154.728	-3886.254	1.621	3
				6006.094	-784.079	-2527.[illegible]		
				7023.743	-55.714	-148.570		
				3323.281	-531.507	-2167.[illegible]36		
51	12	16	10	199.210	710.419	-1570.565	5.571	3
				3810.565	-1118.425	-1226.[illegible]		
				4.069	225.156	3[illegible]		
				1337.954	-56.950	-330.509		
52	5	9	11	3746.187	-2416.415	-2311.[illegible]	2.407	3
				3015.412	-1453.093	2307.114		
				-3305.159	154.568	-228.333		
				1152.147	-1251.506	149.[illegible]40		
53	11	15	9	6751.621	316.348	-28.427	2.560	3
				2153.901	226.647	2458.323		
				-1929.574	-3457.732	1255.563		
				2308.649	-984.246	1223.526		
54	9	14	15	-6666.887	-2591.633	-4305.745	1.364	3
				2094.780	-80.909	-805.[illegible]		
				3249.926	-434.053	-1332.126		
				-240.729	-871.156	-2009.255		
55	15	20	14	391.171	-287.925	691.757	10.000	3
				-2416.722	62.595	356.534		
				166.510	52.752	553.133		
				-619.747	-57.559	733.332		
56	14	18	20	3054.051	57.169	-555.223	6.666	3
				5110.092	-86.326	-449.425		
				-4531.056	-48.579	-738.514		
				1211.029	-25.932	-574.597		
58	13	17	19	17435.952	-2756.263	-263.177	1.341	3
				18266.055	3009.321	-491.549		
				-17634.510	-75.035	237.[illegible]79		
				6022.034	46.008	-172.[illegible]32		
5	19	27	21	27082.771	2119.202	-4922.[illegible]52	.889	1
				37769.434	-3159.936	-3363.[illegible]36		
				26632.673	-536.960	368.[illegible]45		
				30494.759	-515.398	-2519.148		
6	21	29	27	3676.151	1800.157	-5524.719	.923	1
				18512.955	-1732.330	-4432.706		
				34207.239	-1276.959	-1352.[illegible]31		
				16798.763	-463.704	-3671.519		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
11	17	25	23	39406.773	2176.600	318.538	1.562	1
				43749.969	-662.777	-1097.759		
				14460.703	-174.951	-1329.151		
				31545.795	459.041	-436.108		
12	23	31	25	29340.573	2328.997	-4193.335	.635	1
				43006.361	-2259.796	-4809.724		
				9299.573	625.549	-1999.932		
				27482.073	231.517	-3087.417		
17	20	26	22	41179.043	731.152	-1445.981	1.278	1
				44542.349	-372.220	-1174.551		
				18752.323	65.460	303.733		
				34524.772	-57.530	-772.130		
18	22	30	26	29470.276	2302.470	-4307.423	.531	1
				45544.279	-2266.552	-5336.540		
				13746.983	323.120	-1459.502		
				29605.540	119.679	-4371.138		
22	16	24	18	45260.492	-272.955	2321.417	1.166	1
				39228.242	97.337	2322.536		
				16667.021	261.607	-405.531		
				33752.155	26.063	1335.514		
23	18	26	24	34623.596	625.012	-574.936	1.517	1
				36637.233	-1422.373	643.737		
				26831.463	-408.002	-1412.394		
				30563.531	-337.541	-447.754		
24	24	32	26	13677.564	1310.731	-1092.552	1.535	1
				21394.005	-240.552	-2147.733		
				29088.018	-665.010	-2342.512		
				21455.254	55.313	-2061.532		
36	17	21	15	-5633.174	945.569	4563.286	1.307	2
				-15328.425	664.175	2577.750		
				5201.450	-151.163	261.293		
				-5653.396	404.527	2487.441		
37	16	20	18	-3873.514	530.407	-533.528	7.352	2
				-1244.766	-292.597	-1293.556		
				1191.569	41.006	-139.495		
				-1368.737	193.159	-644.730		
38	18	22	20	-5445.476	103.552	-54.542	7.228	2
				-5433.009	-21.219	-208.587		
				2582.059	100.385	471.546		
				-2765.560	63.073	49.535		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LEG
39	20	23	22	-1209.162	-166.245	149.534	10.000	2
				-1052.364	258.077	70.550		
				-572.522	-56.233	365.427		
				-1377.950	5.450	[illegible]92.037		
40	22	25	23	-2618.253	-367.751	2055.724	1[illegible].000	2
				-3673.054	-66.249	642.035		
				1723.515	245.553	-8.055		
				-1559.492	-62.516	299.052		
41	23	27	25	-2550.500	-630.273	1658.435	2.306	2
				-4450.015	645.163	2250.055		
				861.545	-735.051	904.493		
				-1247.057	-175.570	1071.411		
42	25	29	27	-140.542	-1555.655	960.526	3.533	2
				-677.234	1152.399	2011.471		
				-6466.765	-575.056	523.065		
				-1005.215	-323.771	1247.062		
43	24	29	26	6274.402	-1468.080	3567.736	1.234	2
				-3466.020	1151.154	-3655.042		
				5532.561	-963.964	-2115.129		
				2112.961	-353.603	2397.756		
44	26	30	28	1751.194	-1761.743	5976.445	1.004	2
				-6551.464	1660.659	3367.349		
				6259.003	603.001	-2346.542		
				-560.461	362.971	1852.717		
45	28	31	30	-1656.673	-1745.554	5176.013	.736	2
				-9260.023	-657.479	1452.152		
				10545.003	1023.547	-3257.460		
				1964.471	131.024	1456.035		
57	20	24	19	-5054.561	1055.752	963.536	5.636	3
				-6751.014	-722.635	1033.264		
				5176.435	275.563	-323.032		
				-2223.133	-113.151	363.097		
59	19	23	17	6270.249	-745.0[illegible]	1410.595	5.619	3
				2755.020	-455.062	533.451		
				-1576.550	176.403	523.477		
				2350.023	519.[illegible]	926.030		
60	17	22	20	1604.437	[illegible]	1753.[illegible]	4.015	3
				444.576	-1[illegible]	950.[illegible]		
				3733.271	-2[illegible]	-1015.[illegible]		
				1659.095	361.[illegible]	2[illegible]		

ELEMENT NO	I	J	K	SIGMA 1	SIGMA 2	TAU 12	MARGIN	LCS
61	23	28	22	280.010	-191.673	313.142	7.054	3
				-2770.337	-7.170	1337.312		
				8007.043	-155.955	-62.214		
				1040.960	-119.204	722.113		
62	22	26	25	2254.771	-367.350	1753.175	3.475	3
				-3764.754	1128.654	2250.209		
				12206.381	-773.352	458.175		
				3565.443	-200.150	1492.142		
63	28	32	26	4763.560	-1609.374	1247.344	4.564	3
				1226.375	1340.524	1444.440		
				-3200.574	101.126	-225.177		
				929.692	-26.741	322.132		
64	21	25	27	14853.550	-2095.191	4217.175	.676	3
				1827.166	1316.390	6422.131		
				-1665.468	-1552.352	-1803.317		
				5016.449	-645.151	2319.336		
65	27	31	25	5168.723	-2112.202	4211.174	1.050	3
				-4125.714	1658.345	4400.131		
				3423.419	177.633	-2902.171		
				1468.611	-25.362	1903.101		
66	25	30	31	-306.393	-1074.214	6334.327	.575	3
				-5030.279	3985.544	2042.110		
				17452.164	1655.421	-3845.170		
				2665.144	1335.563	1676.136		

MARGIN OF SAFETY IN EACH LEG = -.01075 .09800 .05220

ULTIMATE LOADS

NX	NY	NXY	MX	MY	MXY	TORQUE	QX	QY
1541	0	154	0	0	0	0	0	0

LEG BUCKLING STRESS
6962 6962 6562